Investigating Security of a Few Schemes Based on Public Primitives

A thesis submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in Computer Science
in the
Cryptology Research Group
Applied Statistics Unit
Indian Statistical Institute

Author: Anik Raychaudhuri
Applied Statistics Unit
ISI Kolkata

Advisor: Dr. Mridul Nandi
Applied Statistics Unit
ISI Kolkata

Date of Submission: 20.07.2023

To my friends and family

Acknowledgement

I begin by thanking all those who have emotionally, mentally and logically helped me to complete my thesis. Their support and friendship have helped and motivated me deeply.

Firstly, I express my sincere gratitude to my advisor, Professor Mridul Nandi, for always guiding and supporting me in my research work. His mentorship has played a crucial role in shaping this thesis.

I extend my sincere thanks to all the faculty in ISI and, in particular, those in the Applied Statistics Unit for providing assistance and support at various stages of my research.

I want to thank my collaborators: Rishiraj da, Ritam da, Avijit da and Arghya for supporting my research.

I express my gratitude to all the research scholars in the cryptology group: Ananada, Suprita, Abishanka, Arghya, Sayantan, Avishek, Samir, Jyotirmoy, Soumya, Biswajit, Amit, Aniruddha, Susanta, Chandranan, Nilanjan da and all others. I have not only benefited from discussing my work with you but also immensely enjoyed our light-hearted chit-chat over tea and snacks. I am immensely grateful to you all for extending your friendship towards me.

I thank the office workers in ISI, especially those in ASU, for always providing me with administrative support.

I want to thank my parents for always standing beside me. Their inspiration helped me greatly when times were not so bright for me. They have always been a massive pillar of support for me.

Lastly, I want to thank all my friends from all the stages in my life. My life has been shaped for the better by your presence in it.

Contents

List of Figures

Abstract

Random oracles are cryptographers' conceptions of what an 'ideal' hash function should be. Put succinctly, a random oracle is a perfectly random function that you can evaluate quickly. Random functions are beautiful not just because the output is random-looking (of course) but also because they're automatically collision-resistant and pre-image resistant. However, we know of nothing in the 'real' world that can approximate them. When cryptographers try to analyse their schemes with random functions, they enter an imaginary fantasy world called the 'random oracle model'.

In 2004, Maurer, Renner, and Holenstein gave us a powerful tool for answering this question. What they showed is that it's always possible to replace functionality A (e.g., a random oracle) with another functionality B (e.g., an ideal compression function) provided that the following rules are satisfied: We can 'construct' something 'like' A out of B. We can 'simulate' something 'like' B using A. An attacker who interacts with constructed A-like thing, B cannot tell the difference (i.e., can't differentiate it) from A, simulated B-like thing.

The notion of indifferentiability is very helpful for investigating the security of cryptographic schemes based on public primitives. The public permutation model is another model that helps to scrutinise the security of schemes based on public primitives. In this model, rather than simulating, the adversary also has direct access to the underlying primitives in the ideal world. This model is generally used to analyse keyed constructions as opposed to unkeyed constructions in indifferentiability.

In this thesis, we look at indifferentiability and related security notions in detail. We look back at the definitions and then look at some constructions that achieve the desired security goals. Specifically:

- We look at the 3-round tweakable random permutation-based cipher introduced by Coron et al. in [37] and improve their security results by a factor of 2.
- We also look at the security of Even-Mansour-based key-alternating ciphers in the public permutation model. We show that 5-round Even-Mansour-based key alternating ciphers achieve beyond birthday security ($2n/3-$bits).
- Finally, we dive deeper into the notion of crooked indifferentiability introduced by Russell et al. in [90]. crooked indifferentiability is a novel concept that can be used to build secure constructions from subverted primitives. Russell et al. showed that the enveloped xor construction is crooked indifferentiable from a random oracle. We found some mistakes in their proofs and then corrected them. We also develop a new technique to analyse crooked indifferentiability and then use it to show the security of the Sponge and the Merkle-Damgård constructions, both of which are easier to implement and less costly in memory uses than the enveloped xor construction.

Keywords— Public Primitives, Indifferentiability, Random Oracle, Hash Function, Tweakable Permutation, Subverted Oracle, Kleptography

List of Publications

This thesis is based on the following works;

[1] Bhaumik, R., Nandi, M., Raychaudhuri, A. Improved indifferentiability security proof for 3-round tweakable Luby–Rackoff. Des. Codes Cryptogr. 89, 2255–2281 (2021). https://doi.org/10.1007/s10623-021-00913-4 [22]

[2] Bhattacharjee, A., Bhaumik, R., Dutta, A. et al. BBB security for 5-round even-Mansour-based key-alternating Feistel ciphers. Des. Codes Cryptogr. 92, 13–49 (2024). https://doi.org/10.1007/s10623-023-01288-4 [14]

[3] Bhattacharyya, R., Nandi, M., Raychaudhuri, A. (2021). Crooked Indifferentiability of Enveloped XOR Revisited. In: Adhikari, A., Küsters, R., Preneel, B. (eds) Progress in Cryptology – INDOCRYPT 2021. INDOCRYPT 2021. Lecture Notes in Computer Science(), vol 13143. Springer, Cham. https://doi.org/10.1007/978-3-030-92518-5_4 [20]

[4] Bhattacharyya, R., Nandi, M., Raychaudhuri, A., "Subversion Resilient Hashing: Efficient Constructions and Modular Proofs for Crooked Indifferentiability," in IEEE Transactions on Information Theory, vol. 69, no. 5, pp. 3302-3315, May 2023, doi: 10.1109/TIT.2023.3238115 [21].

Chapter 1

Introduction

1.1 General Discussion

Cryptology is the science of secure communications. Cryptography creates ways to communicate messages between authorised parties so that their meaning remains hidden from any eavesdropper. A cipher is a cryptographic algorithm. A plaintext is an unencrypted message. Encryption converts the plaintext to a ciphertext. Decryption turns a ciphertext back into a plaintext. Confidentiality, integrity, authentication, and non-repudiation Cryptography can provide schemes that are secure for different practical uses. It is important to note that it does not directly provide availability. Cryptography can also provide authentication (proving an identity claim). Additionally, cryptography can provide non-repudiation, which is an assurance that a specific user performed a specific transaction and that the transaction did not change. The two must be tied together. Proving that you signed a contract to buy a car is not useful if the car dealer can increase the cost after you signed the contract. Non-repudiation means the individual who performed a transaction, such as authenticating to a system and viewing personally identifiable information (PII), cannot repudiate (or deny) having done so afterwards.

Cryptographic substitution replaces one character for another, which creates confusion. Permutation (also called transposition) provides diffusion by rearranging the characters of the plaintext, anagram-style. “ATTACKATDAWN” can be rearranged to “CAAKDTANTATW,” for example. Substitution and permutation are often combined. Although these techniques were used historically (the Caesar cipher is a substitution cipher), they are still used in combination while designing modern ciphers, such as the Advanced Encryption Standard (AES) [41].

Strong encryption destroys patterns. If a single bit of plaintext changes, the odds of every bit of resulting ciphertext changing should be 50/50. Any signs of non-randomness may be used as clues to a cryptanalyst, hinting at the underlying order of the original plaintext or key.

So, we can see that we can view cryptography as a series of games or challenges played between two parties, where one party is challenged to keep some information secret while the other party tries to gain some knowledge about it. Cryptographic constructs are greatly useful tools which can be used to fulfil various kinds of security goals. On the other hand, just empirical evidence is often not satisfactory to guarantee the security goals would be fulfilled. So, various kinds of security notions have been developed that correspond to various security aims. One of the more important security notions is that of indifferentiability. In this thesis, we will discuss indifferentiability and some other related security notions. We will investigate whether some practical constructions fulfill those security notions or not, and as a by-product, propose constructions that can be used in various practical scenarios.

Indistinguishability. Broadly speaking, there are two approaches to designing cryptographic systems. The low-level approach is to design a circuit from scratch, say a block cipher like AES [41] or a compression function like Secure Hash Algorithm 1 (SHA1) [1], and claim that it has certain cryptographic properties; such claims are not mathematically provable and are usually accepted by the community only after the design survives prolonged rigorous scrutiny from competent cryptanalysts. The high-level approach begins with a previously-designed primitive and uses it as a black box in a *mode of operation* to build a new cryptosystem on top of it, with an extended domain or a completely different security goal, and an accompanying mathematical reduction-based proof of its security based on some axiomatic idealised property of the underlying primitive.

The most common type of reduction proof is that of indistinguishability, where one shows that as long as certain given black-box assumptions on the underlying primitives hold good, the computational distance between the construction and an ideal random system with the claimed security is low.

Public Primitives. Usually, in the setting of indistinguishability, the primitives used as building blocks for the cryptographic schemes are considered to be secret. In reality, this is often not the case, and the primitives might be accessed publicly. Many real-world cryptosystems depend on such primitives. Public Permutation-based design of cryptographic constructions has seen an increasing number of applications in different symmetric key designs. A notable example is the Sponge design. It has been used in several lightweight authenticated encryption submissions to the NIST lightweight standardised competition. Indifferentiability results tell us that, instead of designing permutations from scratch, the Feistel layout can generically convert a block cipher (modelled to be an ideal cipher) to an ideal permutation. This has been used in Oribatida, one of the second-round NIST lightweight candidates [15]. The permutation SimP is a Feistel-based permutation based on the Simon block cipher [7].

Indifferentiability Framework. The seminal work of Maurer et al. [67] on the *indifferentiability* of random systems introduced a new framework for security proofs when the primitives used as building blocks can be publicly accessed. Suppose f is a given public primitive, and $\mathcal{C}^f$ is a construction with the same claimed security as that of an ideal primitive $\mathcal{P}$. (Following a commonly used notational convention, we use the f in superscript to denote that $\mathcal{C}$ has oracle access to f.) Indifferentiability formalizes a necessary and sufficient condition for the construction $\mathcal{C}^f$ to securely replace its ideal counterpart $\mathcal{P}$ in a wide range of environments: to wit, there exists a simulator $\mathcal{S}$, such that the systems $(\mathcal{C}^f, f)$ and $(\mathcal{P}, \mathcal{S}^{\mathcal{P}})$ are indistinguishable, even when the distinguisher has access to f. According to the composition theorem proved by Maurer et al. [67], if $\mathcal{C}$ is indifferentiable from $\mathcal{P}$, then $\mathcal{C}^f$ can securely replace $\mathcal{P}$ in arbitrary (single-stage, [86]) contexts. Thus, proving that $\mathcal{C}$ is indifferentiable from $\mathcal{P}$ amounts to proving that all security properties implicit in $\mathcal{P}$ also hold for $\mathcal{C}^f$. This approach has been successfully applied to the analysis of many symmetric cryptographic constructions in various ideal-primitive models ([36], [13], [51], [42]).

Public primitive Based Indistinguishability. The theme of indifferentiability is trying to answer the question of what happens if the adversary has access to the primitive upon which the cryptographic construction is built. Public primitive-based indistinguishability is similar in idea, but the difference is, in this case, even in the ideal world the adversary has access to the public primitives instead of the simulator. A common difference with indifferentiability is that Indifferentiability is, in general, discussed in the case of unkeyed construction, while Public primitive-based indistinguishability is discussed in the case of keyed constructions.

Indifferentiability vs. Indistinguishability. Informally, two systems f and g are said to be indistinguishable if no (efficient) algorithm, interacting with either f or g, can determine with a probability significantly better than random which of f and g it is interacting with. The notion of *reducibility* is directly based on indistinguishability. A system f is said to be reducible to G if the system G can be used to construct a new system $\mathcal{B}^g$ which is indistinguishable from f. Again, reducibility is useful for cryptographic security proofs: if f is reducible to g, then, for any construction $\mathcal{C}^f$ using f as a component, there is another construction based on G, namely $\mathcal{C}^{\mathcal{B}^H}$, having the same functionality and, in particular, providing the same security as $\mathcal{C}^f$.

However, these considerations are all subject to the assumption that each component (or primitive) a cryptographic construction is based on is a resource belonging to one specific party that has exclusive access to it, i.e., all other entities are unable to directly influence the component's behaviour or obtain any information about its randomness. In practice, we often have to account for the scenarios where the underlying resources are publicly accessible, and using that it is possible to mount an attack on the cryptographic scheme. For example, while for each party the output of a random oracle $\mathcal{R}$ is indistinguishable from the output of a local random function $\mathcal{R}_0$, the security of a construction based on $\mathcal{R}_0$ might be lost when replacing this component by $\mathcal{R}$.

These shortcomings of the notion of indistinguishability motivate the study of the indifferentiability of cryptographic constructions. Indifferentiability allows for the same general statements about the security of cryptosystems as the conventional definitions. Thus, if a component f is indifferentiable from g, then the security of any cryptosystem $\mathcal{C}^g$ based on g is not affected when replacing g by f. Moreover, if f is not indifferentiable from g, this implies the existence of a construction $\mathcal{C}$ for which these components are not interchangeable, i.e., $\mathcal{C}^g$ is secure but becomes insecure if g is substituted by f.

Crooked Indifferentiability. Suppose we have a hash function or random oracle that responds incorrectly on some inputs or might even be adversarially corrupted. It is interesting to ask if we can securely use them. Specifically, given a function $\tilde{h}$ which has been drawn from a distribution that agrees with a uniform function in most places, we would like to be able to produce a corrected version that appears uniform to adversaries with a polynomially bounded number of queries. This model is partially motivated by the traditional study of "program checking and self-correcting". The goal of this theory is to transform a program that is faulty (i.e. answers incorrectly) only at a small fraction of inputs (modelling an evasive adversary) to a program that is correct at all points with overwhelming probability. The crooked Indifferentiability setting introduced by Russell et al. intuitively adapts this classical theory of self-correction to the study of "self-correcting a probability distribution." Crooked Indifferentiability is conceptually similar to Indifferentiability, but with a caveat that the simulator has access to the subverted primitive.

1.2 Indifferentiability.

In this section, we discuss a bit more about indifferentiability and mention some related works.

1.2.1 Adversarial Games.

In general, cryptological security questions are proposed in the form of an adversarial game. The game is usually played between two parties, a challenger $\mathcal{C}$, and an adversary $\mathcal{A}$. $\mathcal{C}$ and $\mathcal{A}$ are modelled as algorithms, which are usually computationally bounded. The challenge is centred on a cryptological construction about which $\mathcal{A}$ has to make certain deductions. They can be something like finding a pre-image of a certain ciphertext, finding a multicollision pair, or in case of indistinguishability and indifferentiability, deducing whether the environment the adversary is dealing with is the proposed cryptological construction, which we usually denote as the real world or an idealised primitive which we generally denote as the ideal world. The adversary $\mathcal{A}$ has information about some previous communications, which are modelled as query-response pairs. The cryptological construct is generally modelled as a black box, i.e. the adversary can observe the input message and the output ciphertext but is usually unaware of the internal states, though there are exceptions to these rules, like in the case of crooked indifferentiability. We assert a certain bound on the amount of information available to the adversary $\mathcal{A}$ before it makes a decision, which is usually called the maximum number of queries available to A. The adversary A tries to achieve his goal by using a certain amount of information or queries, while the challenger tries to maximise the number of queries for the adversary to fulfil his goal. In general, cryptological constructs have an input size of n-bits of $\{0, 1\}$ binary strings, i.e. the inputs come from the product of n copies of $\{0, 1\}$, where n is a positive integer. If the adversary needs at least $\mathcal{O}(2^{\frac{n}{2}})$ queries to reach his goal then the construction is called birthday secure while needing more queries than that leads a construction to be beyond birthday secure.

1.2.2 Indifferentiability in Terms of Adversarial Games

In the indifferentiability set-up, the adversary is asked to distinguish between two set-ups, which we usually call as the real world and the ideal world. The real world consists of an ideal primitive f and a cryptological construction $\mathcal{P}$ which can interact with f. The ideal world consists of an ideal primitive $\mathcal{F}$ and an algorithm$\mathscr{S}$ which can interact with $\mathcal{F}$. The adversary interacts with either the real world or the ideal world, and after gaining responses to all its queries, outputs either the bit 1 or 0. The distinguishing advantage of the adversary is defined as the absolute difference between the adversary $\mathcal{A}$ outputting 1 in the real world and the ideal world. If this advantage is negligible, we say that $(\mathcal{F}, \mathscr{S})$ is indifferentiable from $(\mathcal{P}, f)$. In this case, using $\mathcal{P}$ in place of $\mathcal{F}$ in a cryptological construction doesn't hurt the security of the construction.

1.2.3 Related Works.

A Feistel network or a Luby-Rackoff network uses a round function, a function which takes two inputs – a data block and a subkey – and returns one output of the same size as the data block. In each round, the round function is run on half of the data to be encrypted, and its output is xored with the other half of the data. This is repeated a fixed number of times, and the final output is the encrypted data. The Feistel network can be designed using tweakable random permutations in place of functions in each round. We denote a Feistel network using 3 rounds of tweakable round permutations as TLR3. Mandal et al. in [65] proved the sequential indifferentiability of TLR3; this is a weaker notion of indifferentiability, where the adversary is not allowed to query the simulator any more after the first construction query. The indifferentiability of classical Feistel networks of varying sizes has been a core area of focus in the study of indifferentiability, [43] and [38] being two recent results. The technique of generalising the structure of Feistel networks has been studied in detail in [58].

The indifferentiability of other well-known constructions is also a popular area of research. Bertoni et al. proved the indifferentiability of the Sponge construction with a random oracle in [13]. Dai et al. in [42] have shown that the 5-round iterated Even-Mansour with a non-idealised key schedule is indifferentiable from an ideal cipher. The indifferentiability of the xor of random permutations was examined in [64], [16], and [62]. Other studies in indifferentiability have covered hash functions [28, 71], permutation-based compression functions [50], tweakable random permutations and pseudorandom permutations [27, 71]. On the more theoretical side, [59, 39] examined the random oracle and ideal cipher models in detail. Constructions based on random permutation are very popular in the crypto-community. The indifferentiability of many such constructions has been deeply analysed. Bertoni et al.in [13] discussed the indifferentiability of Sponge construction based on random permutations. The indifferentiability of constructions based on tweakable random permutations has been discussed in detail in [37] and [22].

1.3 Discussion on Crooked Indifferentiability

Blackbox Reduction and Kleptographic Attack. Many of the modern cryptographic constructions are analysed in a modular and inherently black-box manner. The schemes or protocols are built on underlying primitives, only exploiting the functionality of the primitives. While analysing the security, one shows a reduction saying, a successful attack on the construction will lead to an attack against the underlying primitive. Unfortunately, this approach completely leaves out the implementation aspects. While the underlying primitive may be well studied, a malicious implementation may embed a trapdoor or other sensitive information that can be used for the attack. Moreover, such implementation may well be indistinguishable from a faithful implementation [100]. These types of attacks fall in the realm of *Kleptography*, introduced by Young and Yung [100]. While the cryptographic community did not consider kleptography as a real threat, the scenario has changed in the past few years. The kleptographic attack has been a real possibility in the post-Snowden world [84]. A line of work has appeared aiming to formalise and provide security against kleptographic attacks [9, 45, 88, 89]. Specifically, in [9], Bellare, Paterson, and Rogaway showed that it is possible to mount algorithm substitution attacks against almost all known symmetric key encryption schemes to the extent that the attacker learns the secret key. Another good example is the Dual-EC tampering attack [29] which led to the withdrawal of a standardised PRG due to a potential backdoor in the implementation. A series of work has been done in recent years formalizing approaches to resist algorithm substitution attacks [48, 8, 70, 44, 45, 88, 89, 2, 4].

Indifferentiability of Hash Functions and Security against Algorithm Substitution Attacks (ASA). Hash functions are ubiquitous in modern cryptography. Hash functions are widely popular as the drop-in replacements of Random Oracles (RO) in cryptographic schemes and protocols. To facilitate this application, the notion of *indifferentiability* from a Random Oracle, introduced by Maurer, Renner, and Holenstein [67], has been established as a mainstream security criterion. Indifferentiability from a Random Oracle implies all security guarantees (like collision resistance) satisfied by a Random Oracle in a single-stage game up to the indifferentiability bound. Starting from the work of Coron, Dodis, Malinaud, and Puniya [36], a plethora of results [28, 18, 19, 17, 69, 73] have been proven to show the indifferentiability of different constructions based on different ideal primitives.

Surprisingly, analysis of secure hash functions against Algorithm Substitution Attacks (ASA) has been scarce. In CRYPTO 2018, Russel, Tang, Yung and Zhou [90] studied the problem of correcting subverted Random Oracles. They introduced the notion of crooked indifferentiability as a replacement for classical indifferentiability for the kleptographic setting. They showed that the Enveloped XOR construction could be proven secure in this framework.

Like classical indifferentiability, the game of crooked indifferentiability challenges the adversary to distinguish between two worlds. In the real world, the adversary has access to the underlying ideal primitive f, and the construction C, which has subroutine access to $\tilde{f}$, the subverted implementation of f.[1] The implementation $\tilde{f}$ on input an element x queries the function (possibly adaptively) at maximum $\tilde{q}$ many points and, based on the transcript, decides the evaluation of x. As the adversary likes the subversion to go undetected, it is assumed that $\tilde{f}$ differs from f only on some negligible fraction (ϵ) of the domain.

In the ideal world, the construction is replaced by a Random Oracle $\mathcal{F}$. The role of f is played by a simulator with Oracle access to $\mathcal{F}$ and the subverted implementation $\tilde{f}$. The job of the simulator is to simulate f in such a way that $(C^{\tilde{f}}, f)$ is indistinguishable from $(\mathcal{F}, S^{\mathcal{F},\tilde{f}})$. To avoid trivial attacks, the framework allows a *public* random string R to be used as the salt in the construction. The string R is fixed after the adversary publishes the implementation but stays the same throughout the interaction. All the parties, including the simulator and the adversary, get R as part of the initialisation input. We note that even in the weaker setting of Random Oracles with auxiliary input, a random salt is required to prove security [35, 49].

The notion of crooked indifferentiability from a Random Oracle and the composition theorem proved in [90] guarantees that a construction proved secure in this framework can be used to replace a Random Oracle in any single-stage game in the kleptographic setting. While popular hash functions are the most natural choice for instantiating the Random Oracle, their suitability is still unknown. We ask, *can the popular hashing modes, for some parameters, achieve this many-fold stronger security notion?*

1.4 Key Alternating Ciphers.

A block cipher is a length-preserving encryption function that takes a k-bit key K and an n-bit message X and outputs an n-bit ciphertext Y. The primary security requirement of a block cipher is its pseudorandomness. Unfortunately, we cannot establish the theoretical soundness of the security of block ciphers. Therefore, researchers have focused on proving the security results of block ciphers by idealising some of their components. In this direction, two popular design approaches of block ciphers have been extensively studied—Feistel networks and Substitution-Permutation networks (SPNs). As of today, the design of almost every block cipher roughly falls into one of the above two categories.

Feistel Scheme. Most of the provable security results for Feistel networks fall under the Luby-Rackoff (LR) framework, in reference to the seminal work by Luby and Rackoff [63], where the round functions of the Feistel scheme are pseudorandom functions which are idealised as being uniformly random (and secret) via the standard hybrid argument. It was shown in [63] that the 3-round Feistel scheme is a pseudorandom permutation. Later on, Patarin [77] proved that the 4-round Feistel scheme yields a strong pseudorandom permutation, which means that the scheme is secure even if the adversary is allowed to make inverse queries to the permutation oracle. Following [77], a long series of works have established either better security bounds for the Feistel scheme with a larger number of rounds [68, 80, 58, 6, 72] or have reduced the complexity of the security of the scheme [78, 92, 74, 75]. Ramzan and Reyzin [85] proved that the $(n/2)$-bit security of the 4-round Feistel scheme holds even if the adversary has black-box access to the two inner functions of the construction. Naor and Reingold [76] showed that the similar security bound holds even if one replaces the first and last round of the 4-round Feistel construction with pairwise independent permutations and even weaker constructions were proven secure in [83]. Gentry and Ramzan [54] showed that the public random permutation of the one-round Even-Mansour (EM) cipher [53] $X \mapsto K_1 \oplus P(X \oplus K_1)$ can be replaced by a four-round

[1] The domain extension algorithms are simple, and the correctness of their implementations is easy to verify.

public Feistel scheme, and the resulting construction is still a strong pseudorandom permutation that achieves $O(2^{n/2})$ security bound.

Patarin [79] proved $(3n/4)$-bit strong pseudorandomness security for the 6-round Feistel scheme with the conjecture of proving better bounds of the construction. In [68], Maurer and Pietrzak have proved that the r-round Feistel scheme is secure up to $2^{n(r-1)/r}$ queries. In [80], Patarin analysed the security of the Feistel scheme with five or more rounds. He showed that the 5-round Feistel scheme is secure against all attacks that make only the forward queries, as long as the number of queries is less than 2^n. Moreover, he has also shown that 6-round Feistel is secure against all attacks that make both forward and inverse queries to the construction as long as the number of queries is limited to 2^n. Hoang and Rogaway [58] studied the beyond-birthday-bound security of generalised Feistel networks. In 2010 [82], Patarin showed $O(2^n/n)$ security bound for four, five, and six rounds of balanced Feistel schemes and proved beyond birthday bound security bound for unbalanced Feistel scheme with $2n$-bit to n-bit contracting round functions. A detailed literature study on the security of the Feistel scheme can be found in [72].

Substitution-Permutation Networks. Earlier provable security results for SPN ciphers dealt with resistance to specific attacks such as differential [23] and linear attacks [66]. Recently, a series of works have studied the ideal key-alternating cipher, a.k.a. the Iterated Even-Mansour (IEM) cipher. Chen and Steinberger [31] proved a tight security bound (where the bound matches the best-known attack on the construction) of $2^{rn/(r+1)}$ for the r-round IEM cipher. In the last couple of years, research has focussed on analysing the security of the IEM cipher with fewer permutations and keys. Chen et al. [30] have shown a $(2n/3)$-bit security bound for the 2-round IEM cipher based on a single permutation and one n-bit key. This result was extended by Wu et al. [99] to three rounds of the IEM cipher based on a single n-bit public random permutation that was shown secure up to $O(2^{3n/4})$ queries. A recent work by Tessaro and Zhang [95] showed the existence of non-trivial distributions of the limited independence of the round key for which the r-round IEM cipher achieves optimal security. Along with the study of the IEM cipher, security of the tweakable IEM cipher, where the tweak is mixed with each round key of the IEM cipher, has also been extensively studied in [33, 34, 52].

Key-Alternating Feistel Cipher. Despite the extensive research along the lines of Luby and Rackoff [63], which has been very generic and covers many possible choices of round functions for the Feistel scheme, a concrete scheme is yet to be established to design a keyed block cipher from some simple key-less primitive (e.g. unkeyed round function). Therefore, to design a keyed block cipher, it remains necessary to design some keyed round functions $F_i(K_i, X)$, a task which, unfortunately, is not known to be easier than designing the keyed block cipher itself. On the other hand, concrete block ciphers following Feistel designs like Data Encryption Standard (DES), GOST, Camellia, LBlock [98], Twine [94] usually employ length-preserving key-less functions in each round by XOR-ing each round-key before applying the corresponding round function. This idea naturally corresponds to the Feistel scheme with round functions instantiated with $F_i(K_i \oplus X_i)$, where F_i is a key-less public round function and therefore, at the i-th round of the Feistel scheme, the intermediate state is updated as

$$(X_L^i, X_R^i) \mapsto (X_R^i, F_i(X_R^i \oplus K_i) \oplus X_L^i),$$

where X_L and X_R are two n-bit halves of the state and X_R^i and X_L^i denotes those halves in the i-th round. This model of Feistel design was named the Key-Alternating Feistel (KAF) cipher by Lampe and Seurin [61]. One can see that two rounds of a KAF cipher can be rewritten as a single-key one-round EM cipher, where the permutation P is a two-round public and unkeyed Feistel scheme. When the round functions of the KAF cipher are uniform random public functions, we refer to it as an *ideal* KAF cipher. Thus, the ideal KAF cipher differs from the usual LR framework in two ways: (a) one, the ideal KAF cipher considers complex round functions (i.e., random function oracles) instead of the keyed round functions in LR framework; (b) two, it considers the simplest keying procedure, namely key-XOR-ing.

However, the security gap between LR and KAF ciphers is non-negligible. The best-known generic key-recovery attacks with complexity 2^{2n} break four rounds LR [77], which is in sharp contrast with six rounds KAF [56]. Moreover, Patarin has shown [82, 72] that six (resp. five) rounds of LR achieve optimal pseudorandom (resp. strong-pseudorandom) security. However, Guo and Wang. [55] have shown a generic distinguishing attack against the r-round KAF cipher using $O(2^{n(r-2)/(r-1)})$ queries, which implies that the n-round KAF cipher achieves optimal security. As a result, the Luby-Rackoff framework fails to capture the structural properties in practical Feistel ciphers, and hence, the KAF is likely to capture well the practical security of Feistel designs.

The theoretical security analysis of ideal KAF ciphers is generally done using the random function model, where one models the key-less round functions F_i as public random functions that can be queried by the adversary in a black-box way and try to establish the indistinguishability of $(\mathsf{KAF}_{\mathbf{K}}^{F_1,F_2,\ldots,F_r}, F_1, F_2, \ldots, F_r)$ from $(P, F_1, F_2, \ldots, F_r)$ in the random function model, where P is a $2n$-bit uniform random permutation and $\mathbf{K} = (K_1, K_2, \ldots, K_r)$ contains r uniformly random n-bit keys. This indistinguishability notion implies that the ideal KAF cipher with a secret random key $\mathbf{K}$ is indistinguishable from a $2n$-bit uniform random permutation P, even if the adversary is given access to the r random round functions $F_1, F_2, \ldots, F_r$. Note that this security model is closely related to the security model used in proving the security of the IEM cipher.

In this direction, the first reported work is by Ramzan and Reyzin [85] who proved the $(n/2)$-bit strong pseudorandom security of the 4-round Feistel scheme even when the adversary has black-box access to the middle two functions of the construction. Gentry and Ramzan [54] showed the $(n/2)$-bit strong pseudorandom security of the one-round EM cipher when its underlying public permutation is replaced by a four-round public Feistel scheme. Lampe and Seurin [61] proved that an r-round ideal KAF cipher achieves security up to $O(2^{tn/(t+1)})$ queries of the adversary, where $t = \lfloor r/3 \rfloor$ in the non-adaptive setting with the adversary prohibited in making inverse queries to the construction, and $t = \lfloor r/6 \rfloor$ in the adaptive setting with the adversary allowed to make bi-directional queries to the construction. More recently, Guo and Wang [55] have shown that a 4-round ideal KAF cipher with a single round function F and four n-bit round keys (K_1, K_2, K_3, K_4) such that K_1, K_4 and $K_2 \oplus K_3$ are all uniform is $(n/2)$-bit secure in the multi-user setting; they have further shown that a 6-round ideal KAF cipher with six independent round functions is $(2n/3)$-bit secure in the multi-user setting as long as the six round keys $(K_1, K_2, K_3, K_4, K_5, K_6)$ are all uniform and adjacent round keys are independent. In a follow-up work of [55], Shen et al. [93] studied a 4-round ideal KAF cipher with an even more optimised key schedule, in which an n-bit master key K is XOR-ed only in the first and last rounds of the cipher and a one-bit rotation is applied on the output of the first layer round function, and proved the $(n/2)$-bit strong pseudorandom security of the construction.

1.5 General Structure of the Thesis.

Here we give a brief overview of the general structure of this thesis. In the second chapter 2 we delve into the formal definitions of indifferentiability, crooked indifferentiability, and also the security of EM-based key alternating ciphers. We also state the Markov inequality [97] and a version of the Bernoulli inequality [96]. We also introduce the famous H-coefficient technique used in this thesis more than once. In Chapter 3 we discuss the indifferentiability of the TLR3 construction introduced by Coron et al. in [37]. TLR3 is a construction that uses three n-bit input size tweakable random permutations to produce a $2n$-bit random permutation. They showed that it reaches $n/2$ bit indifferentiability security. Here we improve the bound to almost n-bit. In the next chapter 4 we discuss the security for 5-round Even-Mansour-based key alternating ciphers. We show that in the public permutation model, the construction is secure up to almost $2n/3$ bits. In the next two chapters, we dive deep into the crooked indifferentiability notion and tackle the problem of correcting subverted random oracles. In chapter 5 we discuss the enveloped xor construction introduced by Russell et al. in [90]. We identify some crucial mistakes in their proof and then go on to rectify them. In chapter 6 we discuss variants of Sponge construction and Merkle-Damgård construction and their crooked indifferentiability. We show that they reach crooked indifferentiability security while using significantly less primitives and public randomness, and solve the main open question raised by Russell et al. in [90].

Chapter 2

Preliminaries

We denote the set $\{1, 2, \dots, m\}$ by $[1..m]$ and the set $\{0, 1, \dots m\}$ by $(m]$. For a set $\mathcal{X}$, $|\mathcal{X}|$ will denote its cardinality. Let $N = \{0, 1, \dots\}$ be the set of natural numbers and $\{0, 1\}^*$ be the set of all binary strings. For a natural number n, we write the n-times Cartesian product of the set $\{0, 1\}$ with itself as $\{0, 1\}^n$, which equivalently denotes the set of all n-bit binary strings. 0^n (resp. 1^n) denotes the concatenation of n 0 bits (resp. n 1 bits). We write $\{0, 1\}^{\geq n}$ to denote the set of all binary strings of length at least n and $\{0, 1\}^* = \cup_{n=0}^{\infty} \{0, 1\}^n$ to denote the set of all binary strings. For any $X \in \{0, 1\}^*$, $|X|$ denotes the bit-length of X. For two binary strings $X, Y \in \{0, 1\}^*$, $X\|Y$ denotes the concatenation of X and Y. For two n-bit binary strings $X, Y \in \{0, 1\}^n$, $X + Y$ denotes the field addition of X and Y, equivalent to their bit-wise XOR. For any $X \in \{0, 1\}^*$, we denote the parsing of X into n-bit blocks as $X_1 \cdots X_r \leftarrow_n X$, where $|X_i| = n$ for all $1 \leq i < r$ and $1 \leq |X_r| \leq n$ such that $X = X_1 \| \cdots \| X_r$. We write $\|X\| = \lfloor |X|/n \rfloor$ to denote the number of blocks in X.

We write $X = (X_1, X_2, \cdots, X_t) \in (\{0, 1\}^n)^t$ to denote a t tuple of n-bit binary strings. Given any such t-tuple of n-bit binary strings $X = (X_1, X_2, \cdots, X_t)$ and for any two integers a, b such that $1 \leq a \leq b \leq t$, we write the subtuple $(X_a, X_{a+1}, \cdots, X_b)$ of length $(b - a + 1)$ as $X[a, b]$. For two integers a, b such that $a \leq b$, we write $[a, b]$ to denote the set $\{a, a + 1, \cdots, b\}$. Moreover, when $a = 1$, we write $[1, b]$ as $[b]$ to denote the set $\{1, \dots, b\}$. We write $\mathsf{MSB}_x(X)$ and $\mathsf{LSB}_x(X)$ to denote the most significant x bits and the least significant x bits of the binary string X respectively. For any two integers a, b such that $a \geq b$, we write $(a)_b$ to denote $a(a - 1)(a - 2) \dots (a - b + 1)$.

We write $x \leftarrow\!\!\$\ S$ to denote the process of choosing x uniformly at random from a set S and independently from all other random variables defined so far.

We write $\mathcal{F}_n$ to denote the set of all functions F from $\{0, 1\}^n$ to $\{0, 1\}^n$ and $\mathcal{P}_n$ to denote the set of all permutations P over $\{0, 1\}^n$. For a positive integer r, we write $\mathbf{F}^r = (F_1, F_2, \dots, F_r) \in (\mathcal{F}_n)^r$ to denote a tuple of r n-bit to n-bit functions. Similarly, $\mathbf{P}^r = (P_1, P_2, \dots, P_r) \in (\mathcal{P}_n)^r$ denotes a tuple of r n-bit permutations. For any two tuples of n-bit binary strings $X = (X_1, X_2, \dots, X_t)$ and $Y = (Y_1, Y_2, \dots, Y_t)$ having length t and for any n-bit to n-bit function F, we write $F(X) = Y$ to denote $F(X_i) = Y_i$ for $i \in [t]$. We say that the pair of n-bit binary string tuples (X, Y) is *function compatible*, if there exists at least one function $F : \{0, 1\}^n \to \{0, 1\}^n$ such that $F(X) = Y$. Note that, for (X, Y) to be a function compatible pair, $X_i = X_j \Rightarrow Y_i = Y_j$. Similarly, for an n-bit permutation P, we write $P(X) = Y$ to denote that $P(X_i) = Y_i$ for $i \in [t]$ and in that case, we say that the pair of $n-$bit binary string tuples (X, Y) is *permutation compatible*, if there exists at least one n-bit permutation P such that $P(X) = Y$. Note that, for (X, Y) to be a permutation compatible pair, $X_i = X_j \Leftrightarrow Y_i = Y_j$. We write $\mathbf{F}^r(X) = Y$ (resp.

$\mathbf{P}^r(X) = Y$) to denote $F_i(X) = Y$ (resp. $P_i(X) = Y$) for $i \in [r]$.

2.1 Bernoulli and Markov Inequalities

We first state and prove two well-known inequalities.

Lemma 1. *Suppose* $\forall i \in \{1, .., k\}$, $0 \leq a_i \leq 1$. *Then*

$$\prod_{i=1}^{k}(1 - a_i) \geq 1 - \sum_{i=1}^{k} a_i.$$

Proof. This is a special case of Bernoulli's inequality [96]. The proof is by induction on k. It is trivial for $k = 1$; suppose it holds for all $k \in [1, \ldots, k_0 - 1]$ for some $k_0 \geq 2$. Then

$$\begin{aligned}\prod_{i=1}^{k_0}(1 - a_i) &= (1 - a_{k_0}) \cdot \prod_{i=1}^{k_0-1}(1 - a_i)\\ &\geq (1 - a_{k_0}) \cdot \left(1 - \sum_{i=1}^{k_0-1} a_i\right)\\ &= 1 - \sum_{i=1}^{k_0} a_i + \left(a_{k_0} \cdot \sum_{i=1}^{k_0-1} a_i\right) \geq 1 - \sum_{i=1}^{k_0} a_i.\end{aligned}$$

Thus, by induction, it holds for all k. □

Markov's Inequality. Suppose Y is a random variable taking value in $\mathbb{R}$, the set of real numbers. Then for all $a \geq 0$,

$$\Pr[Y \geq a] \leq \frac{\mathbb{E}(Y)}{a}.$$

This is a well-known result [97], and we include here a short proof for the case when Y is a discrete random variable. From the definition of expectation,

$$\mathbb{E}(Y) = \sum b \cdot \Pr[Y = b] \geq a \cdot \sum_{b \geq a} \Pr[Y = b] = a \cdot \Pr[Y \geq a].$$

Thus,

$$\frac{\mathbb{E}(Y)}{a} \geq \Pr[Y \geq a].$$

□

2.1.1 Security Notions

We use the term *oracle* to denote an interface which provides an adversary with a black-box access to a hidden function. An oracle F providing access to a hidden function ϕ will receive a query x and respond with $\phi(x)$; we simply denote this response as $F(x)$, and say that F provides *oracle access* to ϕ; wherever there is no scope for confusion, we shall refer to F and ϕ interchangeably. We will also use the standard notation F^G to denote that the oracle F has oracle access to the oracle G.

Adversaries . An *adversary* $\mathcal{A}$ is an algorithm possibly with access to oracles $\mathcal{O}_1, \ldots, \mathcal{O}_k$ denoted by $\mathcal{A}^{\mathcal{O}_1,\ldots,\mathcal{O}_k}$. The adversaries considered in this paper are computationally unbounded. The complexities of these algorithms are measured solely by the number of queries they make. An algorithm $\mathcal{A}$ having access to an oracle is called a q-query algorithm if it makes at most q queries to its oracle. Similarly, an oracle algorithm having access to two oracles is called a (q_1, q_2)-query algorithm if it makes at most q_1 and q_2 queries to its first and second oracles, respectively. Adversarial queries and the corresponding responses are stored in a transcript $\boldsymbol{\tau}$.

Definition 1. *Distinguishing Advantage. For two k-tuple of oracles $F^k = (F_1, F_2, \ldots, F_k)$, $G^k = (G_1, G_2, \ldots, G_k)$ and an adversary $\mathcal{A}$, a* distinguishing game *is played as follows: the challenger picks a bit b at random and gives $\mathcal{A}$ access to either F^k or G^k. $\mathcal{A}$ makes a bounded number of adaptive queries and outputs a bit $b \in \{0, 1\}$. We define the* advantage *of an adversary $\mathcal{A}$ at distinguishing F^k from G^k as*

$$\Delta_{\mathcal{A}}(F^k \ ; \ G^k) = \left|\Pr[\mathcal{A}^{F_1,F_2,\ldots,F_k} = 1] - \Pr[\mathcal{A}^{G_1,G_2,\ldots,G_k} = 1]\right|.$$

We will use the word *game* a bit more flexibly to refer to any scenario where the adversary interacts with a k-tuple of oracles. Then, we can think of the distinguishing game described above as the adversary trying to distinguish between two games: one against (F^k) and the other against (G^k). Let us denote the adversary $\mathcal{A}$'s interaction with F^k as $\mathcal{G}$ and that with G^k as $\mathcal{G}'$. We denote by For an adversary $\mathcal{A}$ trying to distinguish between two games $\mathcal{G}$ and $\mathcal{G}'$ we denote the distinguishing advantage of $\mathcal{A}$ as $\Delta_{\mathcal{A}}\left[\mathcal{G}, \mathcal{G}'\right]$, defined as

$$\Delta_{\mathcal{A}}\left[\mathcal{G}, \mathcal{G}'\right] = \left|\Pr[\mathcal{A}^{\mathcal{G}} \text{ returns } 1] - \Pr[\mathcal{A}^{\mathcal{G}'} \text{ returns } 1]\right|.$$

We note that the advantage, being a distance, is subject to the triangle inequality, i.e., for three games $\mathcal{G}$, $\mathcal{G}'$ and $\mathcal{G}''$,

$$\Delta_{\mathcal{A}}\left[\mathcal{G}, \mathcal{G}''\right] \leq \Delta_{\mathcal{A}}\left[\mathcal{G}, \mathcal{G}'\right] + \Delta_{\mathcal{A}}\left[\mathcal{G}', \mathcal{G}''\right].$$

Indifferentiability. An algorithm C with oracle access to an ideal primitive f is said to be (t, q, ϵ)-indifferentiable from an ideal primitive $\mathcal{P}$ if there exists a simulator $\mathscr{S}$ with oracle access to $\mathcal{P}$ and running time at most t, such that for any adversary $\mathcal{A}$ making at most q oracle queries in all, it holds that

$$\Delta_{\mathcal{A}}\left[(\mathcal{P}, \mathscr{S}^{\mathcal{P}}), (C^f, f)\right] \leq \epsilon.$$

We simply call C^f indifferentiable from $\mathcal{P}$ if t is bounded above by some polynomial in q and ϵ is a negligible function of q.

The role of the simulator is not only to simulate the behaviour of f but also to remain consistent with the behaviour of $\mathcal{P}$. Note that the simulator does not see the queries made directly to $\mathcal{P}$, although it can query $\mathcal{P}$ whenever needed. We think of the game against $(\mathcal{P}, \mathscr{S}^{\mathcal{P}})$ as the *ideal world*, and the one against (C^f, f) as the *real world* (since $\mathcal{P}$ is an ideal primitive, while C is usually a real cryptosystem).

2.2 Definition for Ideal Primitives.

Random Oracles. Random Permutations. A (probabilistic) function $f : X \to Y$ with Y finite is said to be a *random oracle* if for each $x \in X$ the value of $f(x)$ is chosen uniformly at random from Y. More precisely, for any distinct $x, x_1, \ldots, x_q \in X$ and any $y, y_1, \ldots, y_q \in Y$,

$$\Pr[f(x) = y \mid f(x_1) = y_1, f(x_2) = y_2, \ldots, f(x_q) = y_q] = \frac{1}{|Y|}.$$

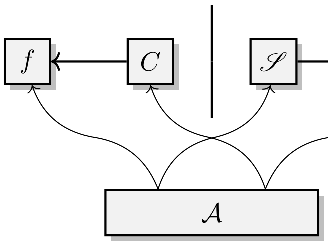


Figure 2.1: The indifferentiability security notion. The real world consists of the construction C and the underlying ideal primitive f. The ideal world consists of the ideal primitive $\mathcal{P}$ and the simulator $\mathscr{S}$. The construction C has oracle access to the underlying primitive P. The simulator $\mathscr{S}$ has oracle access to $\mathcal{F}$. The distinguisher $\mathcal{A}$ interacts either with the real world or with the ideal world.

A function $\pi : X \to X$ with X finite is said to be a *random permutation* if for any distinct $x, x_1, \ldots, x_q \in X$ and any distinct $y, y_1, \ldots, y_q \in X$,

$$\Pr[\pi(x) = y \mid \pi(x_1) = y_1, \pi(x_2) = y_2, \ldots, \pi(x_q) = y_q] = \frac{1}{|X| - q}.$$

A random oracle $f : X \to Y$ can be viewed as a function sampled uniformly from the set of all functions from X to Y. A random permutation is similar to a random oracle except that it is a *permutation* (i.e., bijective and with $Y = X$); one can thus view a random permutation $\pi : X \to X$ as a permutation chosen uniformly at random from the set of all permutations on X.

Tweakable Permutations and Ideal Ciphers. A function $\Phi : \mathcal{T} \times \mathcal{D} \to \mathcal{D}$ is said to be a *tweakable random permutation* if the marginal $\Phi(t, \cdot) : \mathcal{D} \to \mathcal{D}$ is an independent random permutation for each $t \in \mathcal{T}$; here t is called the tweak. Identical in definition but generally used in very different contexts is the *ideal cipher*: a function $E : \mathcal{K} \times \mathcal{D} \to \mathcal{D}$, such that for each key $k \in \mathcal{K}$, $E(k, \cdot) : \mathcal{D} \to \mathcal{D}$ is an independent random permutation. An ideal cipher can be seen as a cipher which can be queried with chosen key-message pairs.

In our proof model for indifferentiability, we consider a construction based on (un-keyed) tweakable random permutations, and allow the adversary to query these tweakable permutations with tweak-message pairs; if we think of the tweak as a key, this is essentially the functionality of an ideal cipher. This allows us to replace the tweakable random permutations with ideal ciphers in the proof.

2.3 H-Coefficient Technique

Suppose a (deterministic) adversary $\mathcal{A}$ is trying to distinguish between two cryptological environments A and A' and the interaction of the adversary with the two environments is modelled as games $\mathcal{G}$ and $\mathcal{G}'$. The transcript $\boldsymbol{\tau}$ is the part of the computation visible to the adversary at the time of choosing its final response. This includes the queries and the responses, and may also include any additional information the oracle chooses to reveal to the adversary at the end of the query-response phase of the game. We say a game $\mathcal{G}$ *yields* $\boldsymbol{\tau}$ to denote the event that $\mathcal{A}$ interacts with $\mathcal{G}$ and obtains $\boldsymbol{\tau}$ as the transcript. Now, for a transcript to be realised, two things need to happen:

- The adversary needs to make the queries listed in the transcript;

- The game needs to make the corresponding responses.

Of these, the former is deterministic; the latter is probabilistic. For instance, consider a transcript for two queries x_1 and x_2, with outputs y_1 and y_2 respectively. This transcript will be realised only when the following four events occur:

- $\mathcal{A}$ begins by querying x_1 (deterministic, depends only on $\mathcal{A}$);
- $\mathcal{G}$ responds to x_1 with y_1 (probabilistic, depends only on $\mathcal{G}$'s randomness after conditioning on first event);
- $\mathcal{A}$, on examining the output y_1, next queries x_2 (deterministic, depends only on $\mathcal{A}$);
- $\mathcal{G}$ responds to x_2 with y_2 (probabilistic, depends only on $\mathcal{G}$'s randomness after conditioning on all earlier events).

Thus, when we talk of the probability of $\mathcal{G}$ yielding a transcript, we are only concerned with the responses of $\mathcal{G}$, with the assumption that the adversary's queries are consistent with the transcript. For any other adversary, this probability is trivially 0. Thus $\Pr[\mathcal{G}$ yields $\boldsymbol{\tau}]$ depends only on $\mathcal{G}$ and $\boldsymbol{\tau}$ and not on the adversary. Now, we state a theorem, due to Patarin [81], that we'll later use in our proofs. The name comes from the original paper, where the (scaled) probabilities of a game yielding a transcript were called H-coefficients.

Theorem 1 (H-Coefficient Technique). *Suppose we can define an event* bad *in game $\mathcal{G}'$, and we call $\boldsymbol{\tau}$ good if it can be obtained from $\mathcal{G}'$ without encountering* bad. *Suppose the following hold:*

- $\Pr[\mathsf{bad}] \leq \epsilon_1$.
- *For any good $\boldsymbol{\tau}$,*

$$\Pr[\mathcal{G} \text{ yields } \boldsymbol{\tau}] \geq (1-\epsilon_2) \cdot \Pr[\mathcal{G}' \text{ yields } \boldsymbol{\tau}].$$

Then for any adversary $\mathcal{A}$ we have

$$\Delta_{\mathcal{A}}[\mathcal{G}, \mathcal{G}'] \leq \epsilon_1 + \epsilon_2.$$

2.4 Security notions and Tools for Crooked Indifferentiability

Class of Functions. $\mathsf{H}_{\mathcal{D},\mathcal{R}}$ denotes the set of all functions from $\mathcal{D}$ to $\mathcal{R}$. $\mathsf{F}_{m,n}$ denotes the set of all functions from $\{0,1\}^m$ to $\{0,1\}^n$. $f : (k] \times \mathcal{D}_f \to \mathcal{R}_f$ denotes a family of k many functions from $\mathcal{D}_f$ to $\mathcal{R}_f$. We often use the shorthand f to denote the family $\{f_1 := f(1,\cdot), \ldots, f_k := f(k,\cdot)\}$ when the function family is given as oracles.

For any tuples of pairs $\tau = ((x_1,y_1),\ldots,(x_{|\tau|},y_{|\tau|}))$ we write $\mathcal{D}(\tau)$ (called domain of τ) to denote the set $\{x_i : 1 \leq i \leq |\tau|\}$. We write $\tau_j = ((x_1,y_1),\ldots,(x_j,y_j))$. We say a function f agrees with τ if for all $(x,y) \in \tau$, $f(x) = y$. For every $x \in \mathcal{D}_f$, $\alpha \in \mathcal{R}_f$, we use $f_{x\to\alpha}$ to denote the following function:

$$f_{x\to\alpha}(y) = \begin{cases} f(y) & \text{if } x \neq y \\ \alpha & \text{if } x = y \end{cases}.$$

Definition 2 (Domain Extension). *Let $\mathcal{D} \supseteq \mathcal{D}_f$. A domain extender C with oracle access to a family of functions $f : [k] \times \mathcal{D}_f \to \mathcal{R}$ is an algorithm that implements the function $\mathsf{H} = C^f : \mathcal{D} \to \mathcal{R}$.*

During the computation of $C^f(M)$, the f query inputs made by C are called the *chaining queries.*

2.4.1 Modeling Subversion Algorithms

We recall the related terms and notations introduced in [90] in our terminologies.

Implementer. A $(q, \tilde{q})$ *implementer* is a q-query oracle algorithm $\mathcal{A}^{\mathcal{O}}$. $\mathcal{A}$ outputs the description of another oracle algorithm $\tilde{F}^{\mathcal{O}}$. The algorithm $\tilde{F}^{\mathcal{O}}$ makes at most $\tilde{q}$ many queries to its oracle. We call $\tilde{F}$ the *implementation*. We let $\tilde{\tau}$ denote the transcript of oracle queries of $\mathcal{A}$. The transcript $\tilde{\tau}$ is hardwired in $\tilde{F}$, and all the $\tilde{q}$ queries made by $\tilde{F}$ are different from $\mathcal{D}(\tilde{\tau})$.

The implementation $\tilde{F}$ is *correct* if for all $f \in \mathsf{H}_{\mathcal{D}_f, \mathcal{R}_f}$ and for all $x \in \mathcal{D}_f$, $\tilde{f}(x) \overset{\text{def}}{=} \tilde{F}^f(x) = f(x)$. A subverted implementation $\tilde{f}$ on input x queries $\alpha_1^{(x)}, \alpha_2^{(x)}, \ldots, \alpha_{\tilde{q}}^{(x)}$, and based on the query-responses outputs $\tilde{f}(x)$. Without loss of generality, we assume $\alpha_1^{(x)} = x$, that is the first query of $\tilde{f}(x)$ is $f(x)$. We use $\alpha \twoheadrightarrow_f \alpha'$ to denote that $\tilde{f}(\alpha)$ queries $f(\alpha')$. Similarly, $\alpha \not\twoheadrightarrow_f \alpha'$, denotes that $\tilde{f}(\alpha)$ does not query $f(\alpha')$. We define the following two sets: (1) $\tilde{Q}_f(x) \overset{\text{def}}{=} \{y \mid x \twoheadrightarrow_f y\}$ and (2) $\overrightarrow{Q}_f(x) \overset{\text{def}}{=} \{y \mid y \twoheadrightarrow_f x\}$. Specifically, $\tilde{Q}_f(x)$ denotes the set $\{\alpha_1^{(x)}, \alpha_2^{(x)}, \ldots, \alpha_{\tilde{q}}^{(x)}\}$. $\overrightarrow{Q}_f(x)$ denotes the set of all points whose (subverted) evaluation queries the point x.

Definition 3 (Crooked Implementer)**.** *A $(q, \tilde{q})$ implementer $\mathcal{A}_1$ is called* ϵ-crooked *for a function family $\mathsf{H}_{\mathcal{D}_f, \mathcal{R}_f}$, if for every $f \in H_{\mathcal{D}_f, \mathcal{R}_f}$, it holds that*

$$\Pr_{\alpha \leftarrow\$ \mathcal{D}_f}[\tilde{f}(\alpha) \neq f(\alpha)] \leq \epsilon$$

where $\tilde{f} \leftarrow \mathcal{A}_1^f$.

Summary. A (crooked) implementation $\tilde{f}$, to compute $\tilde{f}(x)$, queries $f(\alpha_1^{(x)}), \ldots, f(\alpha_{\tilde{q}}^{(x)})$ on $\tilde{q}$ many distinct points ($\alpha_1 = x$) and its decision of whether to subvert $f(\alpha)$ depends on this transcript and the hardwired string $\tilde{\tau}$. For an ϵ-crooked implementation, for each $f \in \mathsf{H}_{\mathcal{D}_f, \mathcal{R}_f}$, for at most ϵ fraction of $x \in \mathcal{D}_f$, $f(x)$ is subverted.

Detection Algorithm. Given an implementation, one may check the algorithm's correctness by comparing the implementation's outputs with a known correct algorithm. More precisely, we sample $\alpha_1, \ldots, \alpha_t \leftarrow\$ \{0,1\}^m$ and then for all $0 \leq i \leq l$, we check whether $\tilde{f}(\alpha_i) = f(\alpha_i)$ holds. If it does not hold, the implementation will be discarded. It is easy to see that for an ϵ-crooked implementation, the subversion would be detected with a probability of at most $t\epsilon$. So for negligible ϵ, this probability would be negligible for all polynomial functions t, and the implementation can survive for further use.

Crooked Distinguisher. A crooked distinguisher is a two-stage adversary; the first stage is a crooked implementer, and the second is a distinguisher.

Definition 4 (Crooked Distinguisher)**.** *We say that a pair $\mathcal{A} := (\mathcal{A}_1, \mathcal{A}_2)$ of probabilistic algorithms* $((q_1, \tilde{q}, \epsilon), q_2)$-crooked distinguisher *for $\mathsf{H}_{\mathcal{D}_f, \mathcal{R}_f}$ if*

(i) $\mathcal{A}_1(r)$ is a ϵ-crooked $(q_1, \tilde{q})$ implementer for $\mathsf{H}_{\mathcal{D}_f, \mathcal{R}_f}$ and

(ii) $\mathcal{A}_2(r, \tilde{\tau}, R)$ is a q_2-query distinguisher where r is the random coin of $\mathcal{A}$, $\tilde{\tau}$ is the advice-string, the transcript of the interaction of $\mathcal{A}_1$ with f, and R is the (randomised) initial vector of the target construction. The random string r and the advice-string $\tilde{\tau}$ are hardwired to $\mathcal{A}_2$, and the random IV R is provided as input.

Crooked Indifferentiability. Now, we state the crooked-indifferentiable security definition (as introduced in [90]) in our notation and terminology. The definition is based on the following two-stage distinguishing game. The ideal primitives f and $\mathcal{F}$ are sampled. The crooked-distinguisher $\mathcal{A}$ (with random string r as the random coins) runs the first phase $\mathcal{A}_1$. The crooked implementer $\mathcal{A}_1$, with oracle access to f, produces a subverted implementation $\tilde{F}$. Then, a uniformly random string R is sampled and published as the IV of the construction C. Finally, $\mathcal{A}_2$ is invoked with an internal random string r, the advice-string $\tilde{\tau}$, and the random IV R as input. In the real world, $\mathcal{A}_2$ interacts with the f (same from the first stage) and the construction $C^{\tilde{f}}(R, \cdot)$. In the ideal world, the simulator S gets the advice-string $\tilde{\tau}$, the initial value R, and black box access to the subverted implementation $\tilde{F}$ as inputs, along with oracle access of a random oracle $\mathcal{F}$. The simulator is aimed to simulate f so that the behaviour of $(f, C^{\tilde{f}})$ is as close as $(S, \mathcal{F})$ to the distinguisher $\mathcal{A}_2$.

Definition 5 (crooked indifferentiability [90])**.** *Let $\mathcal{F}$ be an ideal primitive and C^f be an IV-based $\mathcal{F}$-compatible oracle construction. The construction C is said to be* $((q_1, \tilde{q}),\ (q_2, q_{\text{sim}}),\ \epsilon, \delta)$**-crooked-indifferentiable from** $\mathcal{F}$ *if there is a q_{sim}-query algorithm S (called simulator) such that for all $((\epsilon, q_1, \tilde{q}), q_2)$-crooked distinguisher $(\mathcal{A}_1(r),\ \mathcal{A}_2(r, \cdot, \cdot))$ for $\mathsf{H}_{\mathcal{D}_f, \mathcal{R}_f}$, we have*

$$\Delta_{\mathcal{A}_2(r,\tilde{\tau},R)}\big((f, C^{\tilde{f}}(R, \cdot))\ ;\ (S^{\mathcal{F},\tilde{F}}(\tilde{\tau}, R), \mathcal{F})\big) \leq \delta \tag{2.1}$$

where $\tilde{\tau}$ is the advice string of $\mathcal{A}_1^f$. R is the random initial value of the construction sampled after the subverted implementation is set.

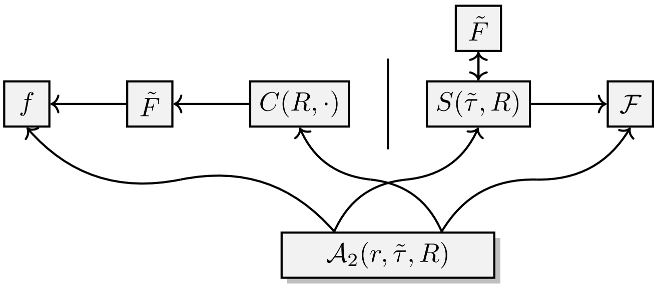


Figure 2.2: The crooked indifferentiability notion. In the first phase of the real world, $\mathcal{A}_1$ interacts with f and returns an oracle algorithm $\tilde{F}$ (which would be accessed by the construction C in the second phase). In the second phase, the random initial value R will be sampled and given to construction C and also to $\mathcal{A}_2$. In the ideal world, the simulator $S^{\mathcal{F}}$ gets the transcript of the first phase as an advice string, black-box access to the subverted implementation $\tilde{F}$ and the initial value R.

Remark 1. *The simulator S gets a black box subroutine access to the algorithm $\tilde{F}$. The simulator can compute $\tilde{F}(x)$ by invoking $\tilde{F}$ with input x and responding to the oracle queries made by $\tilde{F}$.*

Convention on Crooked Distinguishers. Note that there is no loss in assuming that both $\mathcal{A}_1$ and $\mathcal{A}_2$ are deterministic (so we skip the notation r) when we consider a computationally unbounded adversary. $\mathcal{A}$ can fix the best internal random coin r for which the distinguishing advantage of $\mathcal{A}_2$ is maximum. As the randomness of $f, \mathcal{F}$, the public IV R, and the internal random coins of S are independently sampled from r, the maximum distinguishing advantage would follow from an averaging argument.

We also assume that $\mathcal{A}_2$ makes all distinct queries distinct from those made by $\mathcal{A}_1$. We skip the notation $\tilde{\tau}$ as an input of $\mathcal{A}_2$ as it is fixed throughout the game. As the advice string is fixed, we consider it part of the transcript. Specifically, the transcript τ_0, view of $\mathcal{A}_2$ at the start of the second stage, is set as the advice string $\tilde{\tau}$.

Chapter 3

Indifferentiability of Tweakable LR3

3.1 Introduction

Coron et al. in TCC 2010 proposed a $2n$-bit permutation by using an n-bit ideal cipher with an n-bit key in a Feistel type structure for three rounds [37]. The authors showed $n/2$-bit indifferentiable security. In this section, we improve their result by showing $(n-2\log n)$-bit indifferentiable security by using more sophisticated counting techniques. This result will help us design ideal permutations using block ciphers and deploy them in permutation-based cryptosystems such as Sponge constructions.

3.1.1 The Original Construction by Coron et al. [37]

Block ciphers have been a reliable source of cryptographic security for decades. The most well-known block ciphers receive dedicated attention from the world's leading cryptanalysts, and the ones that stand the test for a reasonable time come to gain a trusted position in the cryptographic community. Since block ciphers usually have a fixed small width, the common approach for extending their domain is through modes of operation. Many wider permutations have been built on top of block ciphers which exhibit strong pseudorandomness among other useful properties. However, these modes of operation are usually not indifferentiable from ideal random permutations, and thus cannot be treated as ideal ciphers upon which we can build more cryptosystems.

Coron et al. [37] first introduced a construction that extends the domain of a block cipher from n-bits to $2n$-bits. They used a tweakable block cipher of width n bits in a three-round Feistel mode with three independent keys and showed that this is indifferentiable from an ideal random permutation over $2n$ bits, and thus can be used as an ideal cipher. A detailed description along with a diagram (Fig. 3.1) can be found in Sec. 3.1.2, pp. 18-18 of this paper. (We'll call this construction TLR3, short for *T*weakable *L*uby-*R*ackoff *3*-round.) Their proof of indifferentiability works for an adversary making $\mathcal{O}(2^{n/2})$ queries, guaranteeing security up to the birthday bound in the input width of the underlying tweakable permutation.

3.1.2 3-round TBC-based Luby-Rackoff

We will study the permutation TLR3, due to Coron et al. [37], $\Psi : \{0,1\}^{2n} \to \{0,1\}^{2n}$ has oracle access to primitives E_1, E_2, and E_3, three un-keyed independent tweakable random permutations $\{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$. Since the width of the underlying primitives is n bits, we'll think of Ψ as accepting and outputting pairs of n-bit blocks. For an encryption query (L, R), $(S, T) := \Psi(L, R)$ is computed through the following steps:

$$\begin{aligned} X &\leftarrow E_1(R, L), \\ T &\leftarrow E_2(X, R), \\ S &\leftarrow E_3(T, X). \end{aligned}$$

Similarly, for a decryption query (S, T), $(L, R) := \Psi^{-1}(S, T)$ is computed through the following steps:

$$\begin{aligned} X &\leftarrow E_3^{-1}(T, S), \\ R &\leftarrow E_2^{-1}(X, T), \\ L &\leftarrow E_1^{-1}(R, X). \end{aligned}$$

As explained at the end of Sec. 2.1.1 on p. 13, we shall treat E_1, E_2, E_3 as ideal ciphers in our proof. Fig. 3.1 illustrates the TLR3 construction.

We denote $E := (E_1, E_2, E_3)$ and write Ψ as Ψ^E to explicitly indicate that Ψ has oracle access to E. *We'll consistently use the letters L, R, X, S, T to indicate the same blocks as in Fig. 3.1.* We also note that a query in either direction is completely described by the 5-tuple $\sigma = (L, R, X, S, T)$, including all internal inputs and outputs of the ideal ciphers. For such a 5-tuple σ and an index $i \in [1..5]$ we'll denote the i-th element of σ by σ_i (e.g., $\sigma_3 = X$).

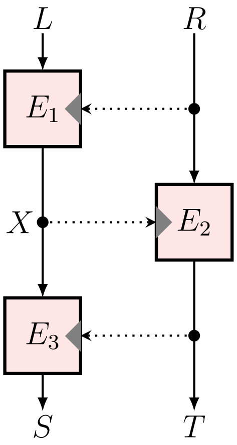


Figure 3.1: TLR3, a permutation-based on 3-round Feistel type mixing using ideal ciphers E_1, E_2 and E_3. The dotted input lines indicate the keys.

3.1.3 Our Contribution

In this chapter, we examine the indifferentiability of Coron et al.'s construction more closely and show that indifferentiability still holds when we allow the adversary to make $\mathcal{O}(2^{n-2\log n})$

queries. This is often found to be the optimal level of security for any construction based on a permutation of width n bits, up to a factor of n.

More specifically, we construct a simulator such that for any adversary making queries to the construction and the simulator not exceeding q queries in all, their differentiating advantage is bounded by $n^2q/2^n + 6nq/2^n + n^2q^2/2^{2n}$, and the simulator makes no more than nq queries to the ideal permutation oracle. (We make the trivial assumption that $n \geq 6$.)

To achieve this better bound, we use a more sophisticated simulator and delicate counting techniques. In particular, when answering inner queries, the simulator in [37] only makes one attempt at sampling a *good* response, while our simulator makes n independent attempts before giving up. We also allow a lot of internal collisions, and only ban collisions with certain properties; this necessitates a lot of careful counting. A detailed overview of our proof technique can be found in Sec. 3.1.4, pp. 19-20.

The main security result of this paper is the following:

Theorem 2. *For the TLR3 construction Ψ with oracle access to a tweakable permutation E of output width $n \geq 6$ bits (both as specified in Sec. 3.1.2, pp. 18-18), there exists a simulator $\mathscr{S}$ such that for any adversary $\mathcal{A}$ making at most q queries, we have*

$$\Delta_{\mathcal{A}}\left[(\pi, \mathscr{S}^{\pi}), (\Psi^E, E)\right] \leq \frac{n^2q}{2^n} + \frac{6nq}{2^n} + \frac{n^2q^2}{2^{2n}},$$

where π is a random permutation over $\{0,1\}^{2n}$. Further, $\mathscr{S}$ makes at most nq queries to π and has running time $\mathcal{O}(nq^2)$.

Hence, Ψ^E is (t, q, ϵ)-indifferentiable from π where $t = \mathcal{O}(nq^2)$ and $\epsilon = n^2q/2^n + 6nq/2^n + n^2q^2/2^{2n}$. (We'll revisit and prove this theorem at the end of the chapter in Subsec. 3.4.4, pp. 36-36.)

We believe this security improvement is a significant leap from the best-known results, especially at a time when permutation-based cryptosystems are gaining quickly in popularity. Our result establishes the TLR3's claim to being a domain-extending ideal cipher on a much firmer footing.

3.1.4 Overview of Proof and Outline of the Chapter.

To prove the indifferentiability of the TLR3 construction Ψ based on a tweakable permutation E from a random permutation π, we first describe a simulator $\mathscr{S}$, and then consider an adversary distinguishing between two games: $\mathcal{G}_0$ with $(\pi, \mathscr{S}^{\pi})$, and $\mathcal{G}_1$ with (Ψ^E, E). (In Ψ^E, we model the tweakable permutation E as an ideal cipher with the tweak as the key.) We consider a hybrid game $\mathcal{G}_{1/2}$ in between, where the random permutation π is replaced with a special random oracle $\widetilde{\pi}$ which answers both input queries and inverse queries (its behaviour is described in detail in Sec. 3.2. The proof reduces to carefully bounding the two advantages $\Delta_{\mathcal{A}}\left[\mathcal{G}_0, \mathcal{G}_{1/2}\right]$ and $\Delta_{\mathcal{A}}\left[\mathcal{G}_{1/2}, \mathcal{G}_1\right]$. We calculate these bounds through separate applications of Patarin's H-Coefficient Technique (Theorem 1, p. 14), and bring everything together to establish the main result.

Sec. 3.2 (pp. 20-23) describes the main simulator $\mathscr{S}$ we use for the proof; it deviates from Coron et al.'s simulator [37] in how it processes the inner queries—where their simulator makes only one attempt to come up with a 'good' response, our simulator makes n attempts before giving up. The pseudocodes for $\mathscr{S}^{\pi}$ can be found in Fig. 3.3,3.4.

Lemma 4 (p. 26) gives a bound for $\Delta_{\mathcal{A}}\left[\mathcal{G}_0, \mathcal{G}_{1/2}\right]$, and Lemma 7 (p. 35) gives a bound for $\Delta_{\mathcal{A}}\left[\mathcal{G}_{1/2}, \mathcal{G}_1\right]$; from these two lemmas, Theorem 2 follows immediately. Proof of Lemma 4 (Sec. 3.3.2, pp. 24-26) is a fairly straightforward application of the H-Coefficient Technique:

we define an event bad^* in $\mathcal{G}_{1/2}$, bound its probability (Lemma 2, p. 25), and show that any good transcript is at least as likely to be yielded by $\mathcal{G}_0$ as by $\mathcal{G}_{1/2}$ (Lemma 3, p. 26); these results combine to establish Lemma 4.

The main challenge lies in proving Lemma 7, i.e., bounding the second advantage $\Delta_{\mathcal{A}}\left[\mathcal{G}_{1/2}, \mathcal{G}_1\right]$; this we do in Sec. 3.4 (pp. 27-36), the longest and the most technically dense section of this work. One crucial problem here is that the adversary can observe multi-collisions in the π queries and use that information to query the simulator. (In [65] this problem does not appear as the adversary can not make any simulator queries after it makes its first construction query.) As in the case of Lemma 4, we define an event bad in $\mathcal{G}_{1/2}$, and split the problem into two parts: bounding the probability of bad (Lemma 5, p. 29) and bounding the ratio of H-Coefficients (Lemma 6, p. 31). In Sec. 3.4.2 (pp. 29-31), through a careful case-by-case counting, we prove Lemma 5. In Sec. 3.4.3 (pp. 31-35) another exercise in careful counting establishes Lemma 6, thus successfully completing the proof of our main result.

3.2 Description of Simulator and $\tilde{\pi}$

$\tilde{\pi}$ represents a specially defined oracle used in the game $\mathcal{G}_1$, which we call a two-sided random oracle. $\tilde{\pi}$ maintains a table Q of its previously answered queries. It takes as input a pair (U, V) and a direction $+$ or $-$, where $+$ indicates a forward query and $-$ indicates a backward query. It first attempts to answer the query from Q: this can be done either if (U, V) has been queried in the same direction before (in which case the pair returned earlier is returned again), or if (U, V) was itself the answer to an earlier query in the opposite direction (in which case the input pair from the aforementioned query is returned). Failing this, it samples an element (W, Z) from $\{0,1\}^{2n}$ uniformly at random, and checks whether (W, Z) is already matched in Q, i.e. if (W, Z) was previously queried in the opposite direction of the present (U, V) query, or was the response to some (U', V') query in the same direction as the (U, V) query. If (W, Z) is found to be *fresh* (not matched before), $\tilde{\pi}$ returns (W, Z) updates Q; otherwise it aborts and returns $\perp$. The pseudocode can be found in Fig. 3.2.

It differs from a lazily sampled random permutation in the fact that new pairs are always sampled uniformly at random from the entire $\{0,1\}^{2n}$, without excluding previously sampled pairs, as we would do for a random permutation. This allows one pair to be matched with multiple pairs, in which case $\tilde{\pi}$ aborts; we treat this as a bad event in the game where we encounter $\tilde{\pi}$. Thus $\tilde{\pi}$ ensures that its table Q remains a partial permutation. We will write $\tilde{\pi}$ and $\tilde{\pi}^{-1}$ to indicate the two-block functions $\tilde{\pi}(+,\cdot,\cdot)$ and $\tilde{\pi}(-,\cdot,\cdot)$ respectively.

In this section we define a simulator $\mathscr{S}$ in preparation for our analysis of the indifferentiability of TLR3. We want the $(\pi, \mathscr{S}^{\pi})$-interdependence to mimic the (Ψ^E, E)-interdependence as closely as possible. This will be our principle when defining the simulator responses.

For each query, $\mathscr{S}$ will first choose a full 5-tuple (L, R, X, S, T) (using the mechanism described below), and then output the required variable from this 5-tuple. There are six types of queries that the adversary can make to the simulator: E_i and E_i^{-1} for $i \in [1..3]$. Of these, $\mathscr{S}$ treats the following pairs of queries identically (except for the output value):

- $E_1^{-1}(R, X)$ and $E_2(X, R)$;
- $E_2^{-1}(X, T)$ and $E_3(T, X)$.

Hence we can define $\mathscr{S}$ by considering four cases, where each case can be identified simply by the unordered pair of query inputs. $\mathscr{S}$ stores the responses to previous queries in a table $\mathcal{H}$, and while answering queries it consults $\mathcal{H}$ as needed; it also maintains a separate table $\mathcal{F}$ of failed attempts (to be described shortly). For convenience we will use the following notation: at any

```
$\widetilde{\pi}(+, L, R)$:
if $(\exists(S,T))((L,R,S,T) \in Q)$
    return $(S,T)$
else
    $(S,T) \leftarrow\$ \{0,1\}^{2n}$
    if $(\exists(L',R'))((L',R',S,T) \in Q)$
        return $\perp$
    else
        $Q \leftarrow Q \cup (L,R,S,T)$
        return $(S,T)$
    endif
endif
```

```
$\widetilde{\pi}(-, S, T)$:
if $(\exists(L,R))((L,R,S,T) \in Q)$
    return $(L,R)$
else
    $(L,R) \leftarrow\$ \{0,1\}^{2n}$
    if $(\exists(S',T'))((L,R,S',T') \in Q)$
        return $\perp$
    else
        $Q \leftarrow Q \cup (L,R,S,T)$
        return $(L,R)$
    endif
endif
```

Figure 3.2: Pseudocode for $\widetilde{\pi}$ on forward and backward queries. Q is initialised as an empty set.

point in the query phase, $E_i(a,\cdot)$ for $i \in [3]$ will denote the set of all c such that $\mathscr{S}$ has set the value c for some $E_i(a,b)$ previously; similarly $E_i^{-1}(a,\cdot)$ for $i \in [1..3]$ will denote the set of all b such that $\mathscr{S}$ has set the value x for some $E_i^{-1}(a,c)$ previously. Finally, $\max\{A,B\}$ will denote the bigger set between A and B in terms of cardinality. With this convention in mind, we now proceed to describe how our simulator works:

$\{\mathbf{L},\mathbf{R}\}$: While answering an $E_1(R,L)$ query $\mathscr{S}$ first checks if for some X,S,T there already exists a 5-tuple (L,R,X,S,T) in $\mathcal{H}$. If yes, then it returns X. Otherwise it sets $(S,T) = \pi(L,R)$. Then it samples X uniformly at random from $\{0,1\}^n \setminus \max(E_1(R,\cdot), E_3^{-1}(T,\cdot))$. $\mathscr{S}$ returns X and stores (L,R,X,S,T) in $\mathcal{H}$. (We mention here that we deviate slightly from the 'common-sense' simulator which would sample X uniformly from $\{0,1\}^n \setminus (E_1(R,\cdot) \bigcup E_3^{-1}(T,\cdot))$; this is done to help simplify the calculations.)

$\{\mathbf{S},\mathbf{T}\}$: This is similar to the $\{L,R\}$ case.

$\{\mathbf{R},\mathbf{X}\}$: Handling this case is a bit trickier. Again $\mathscr{S}$ begins by trying to find L,S,T such that (L,R,X,S,T) is already in $\mathcal{H}$. If it succeeds, it returns L or T depending on whether the query was $E_1^{-1}(R,X)$ or $E_2(X,R)$. Otherwise, $\mathscr{S}$ samples L uniformly at random from $\{0,1\}^n \setminus E_1^{-1}(R,\cdot)$ and queries π with (L,R) to get (S,T). If the tuple (L,R,X,S,T) doesn't violate the tweakable random permutation property (i.e., if $T \notin E_2(X,\cdot)$), $\mathscr{S}$ adds it to $\mathcal{H}$ and returns L or T as needed. Otherwise, it stores (L,R,S,T) in $\mathcal{F}$ and repeats the process starting with re-sampling L. If after n attempts it remains unsuccessful, $\mathscr{S}$ aborts and returns $\perp$.

$\{\mathbf{X},\mathbf{T}\}$: This is similar to the $\{R,X\}$ case.

The pseudocode for the behaviour of $\mathscr{S}^{\pi}$ can be found in Fig. 3.3, Fig. 3.4 (p. 22, p. 23). In the pseudocode we denote $\{0,1\}^n \setminus A$ as A^c. For notational brevity we use the lookup functions $\mathscr{S}_{\mathcal{H}}^{(a,b)}(U,V)$, defined as follows: for an input (U,V) and indices $a,b \in [1..5]$ $\mathscr{S}$ checks if there already exists a 5-tuple σ in $\mathcal{H}$ such that $\sigma_a = U$ and $\sigma_b = V$; if yes then $\mathscr{S}_{\mathcal{H}}^{(a,b)}(U,V)$ returns the other 3 values in σ as an ordered triple; otherwise it returns 0.

```
$E_1(R, L)$
if $\mathscr{S}_{\mathcal{H}}^{(1,2)}(R, L) = (X, S, T)$
  return $X$
else
  $(S, T) \leftarrow \pi(L, R)$
  $X \leftarrow\$ \max(E_1(R, .), E_3^{-1}(T, .))^c$
  add $(L, R, X, S, T)$ to $\mathcal{H}$
  return $X$
endif
```

```
$E_3^{-1}(T, S)$
if $\mathscr{S}_{\mathcal{H}}^{(4,5)}(T, S) = (X, L, R)$
  return $X$
else
  $(L, R) \leftarrow \pi^{-1}(S, T)$
  $X \leftarrow\$ \max(E_1(R, .), E_3^{-1}(T, .))^c$
  add $(L, R, X, S, T)$ to $\mathcal{H}$
  return $X$
endif
```

```
$E_1^{-1}(R, X)$
if $\mathscr{S}_{\mathcal{H}}^{(2,3)}(R, X) = (L, S, T)$
  return $L$
else
  for $j = 1..n$ do
    $L_j \leftarrow\$ E_1^{-1}(R, .)^c$
    $(S_j, T_j) \leftarrow \pi(L_j, R)$
    if $\mathscr{S}_{\mathcal{H}}^{(3,5)}(X, T_j) = 0$
      add $(L_j, R, X, S_j, T_j)$ to $\mathcal{H}$
      return $L_j$
    else
      add $(L_j, R, S_j, T_j)$ to $\mathcal{F}$
    endif
  endfor
  return $\perp$
endif
```

```
$E_2(X, R)$
if $\mathscr{S}_{\mathcal{H}}^{(2,3)}(R, X) = (L, S, T)$
  return $T$
else
  for $j = 1..n$ do
    $L_j \leftarrow\$ E_1^{-1}(R, .)^c$
    $(S_j, T_j) \leftarrow \pi(L_j, R)$
    if $\mathscr{S}_{\mathcal{H}}^{(3,5)}(X, T_j) = 0$
      add $(L_j, R, X, S_j, T_j)$ to $\mathcal{H}$
      return $T_j$
    else
      add $(L_j, R, , S_j, T_j)$ to $\mathcal{F}$
    endif
  endfor
  return $\perp$
endif
```

Figure 3.3: Pseudocodes for $\mathscr{S}^{\pi}$. The rest of the figure is continued in 3.4.

```
$E_2^{-1}(X,T)$
if $\mathscr{S}_{\mathcal{H}}^{(3,5)}(X,T) = (L,R,S)$
  return $R$
else
  for $j = 1..n$ do
    $S_j \leftarrow\!\!\$\ E_3(T,\cdot)^c$
    $(L_j, R_j) \leftarrow \pi^{-1}(S_j, T)$
    if $\mathscr{S}_{\mathcal{H}}^{(2,3)}(R_j, X) = 0$
      add $(L_j, R_j, X, S_j, T)$ to $\mathcal{H}$
      return $R_j$
    else
      add $(L_j, R_j, S_j, T)$ to $\mathcal{F}$
    endif
  endfor
  return $\perp$
endif
```

```
$E_3(T,X)$
if $\mathscr{S}_{\mathcal{H}}^{(3,5)}(X,T) = (L,R,S)$
  return $S$
else
  for $j = 1..n$ do
    $S_j \leftarrow\!\!\$\ E_3(T,\cdot)^c$
    $(L_j, R_j) \leftarrow \pi^{-1}(S_j, T)$
    if $\mathscr{S}_{\mathcal{H}}^{(2,3)}(R_j, X) = 0$
      add $(L_j, R_j, X, S_j, T)$ to $\mathcal{H}$
      return $S_j$
    else
      add $(L_j, R_j, S_j, T)$ to $\mathcal{F}$
    endif
  endfor
  return $\perp$
endif
```

Figure 3.4: Pseudocodes for $\mathscr{S}^{\pi}$. The lookup functions $\mathscr{S}_{\mathcal{H}}^{(a,b)}$ are as defined in Sec. 3.2 (p. 20).

3.2.1 Efficiency of $\mathscr{S}$

To answer each simulator query of the adversary $\mathcal{A}$, $\mathscr{S}$ makes at most n queries to π and for each of them spends at most $\mathcal{O}(q)$ amount of time to determine whether its query provides a valid answer or not. As there are at most q queries made to the simulator, the running time of $\mathscr{S}$ is $\mathcal{O}(nq^2)$ and $\mathscr{S}$ makes at most nq queries to π.

3.3 Introducing the Hybrid Game $\mathcal{G}_{1/2}$

To facilitate our analysis of the indifferentiability of TLR3, we introduce a hybrid game between the ideal world and the real world. Recall the definition of $\widetilde{\pi}$ (Sec. 3.2. Consider an adversary $\mathcal{A}$ playing the following games:

$\mathcal{G}_0$: This game corresponds to the ideal world, where $\mathcal{A}$ interacts with $(\pi, \mathscr{S}^{\pi})$.

$\mathcal{G}_{1/2}$: In this game we replace π with $\widetilde{\pi}$, so $\mathcal{A}$ interacts with $(\widetilde{\pi}, \mathscr{S}^{\widetilde{\pi}})$.

$\mathcal{G}_1$: Finally, this game corresponds to the real world, where $\mathcal{A}$ interacts with (Ψ^E, E).

Fig. 3.5 shows this sequence of games. To prove our intended security bound, we tackle two problems separately: bounding the distinguishing advantage between $\mathcal{G}_0$ and $\mathcal{G}_{1/2}$ and bounding the distinguishing advantage between $\mathcal{G}_{1/2}$ and $\mathcal{G}_1$. We devote the rest of this section to the former and defer the latter to Sec. 3.4. We begin by describing the games in more detail, including descriptions of oracle behaviour and transcripts.

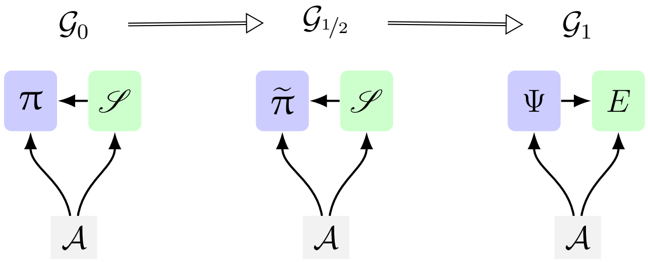


Figure 3.5: Sequence of Games Transforming from Ideal to Real. (Arrows indicate oracle access.)

3.3.1 Transcripts and Adversary Restrictions

Consider an adversary $\mathcal{A}$ playing one of the games $\mathcal{G}_0, \mathcal{G}_{1/2}, \mathcal{G}_1$. We stipulate

- that $\mathcal{A}$ is not allowed to repeat a query;
- that $\mathcal{A}$, for each simulator query, receives the entire 5-tuple (L, R, X, S, T) corresponding to that query;
- that $\mathcal{A}$, after completing all its queries and getting the answers, feeds all its Ψ (resp. π) queries to E (resp. $\mathscr{S}$); E or $\mathscr{S}$ can then calculate all the intermediate X values for those queries as well, and these values are released to $\mathcal{A}$;
- that at the end of the query phase, $\mathscr{S}$ extends all the 4-tuples in $\mathcal{F}$ to 5-tuples by treating them as $\{L, R\}$ queries and reveals these 5-tuples to $\mathcal{A}$ (as for $\mathcal{G}_1$, we describe below a way for E to simulate these failed queries).

We note that these conditions in no way restrict $\mathcal{A}$'s power: it clearly cannot get any extra information by repeating queries; it is free not to use the extra information it gains when it receives the entire 5-tuple instead of its specific query; and finally, since $\mathcal{A}$ has already seen the answers to its queries beforehand and $\mathscr{S}$ does not get to know the π queries of $\mathcal{A}$ before it answers all the simulator queries, the extra information revealed at the end to $\mathcal{A}$ cannot diminish its power. We can thus view the transcript as a set of q^* 5-tuples.

Sampling Failed Attempts in Real World. On an $\{R, X\}$ query, E samples an L uniformly at random from $E_1^{-1}(R, \cdot)$, sets $(S^*, T^*) = \Psi(L^*, R)$, and checks if $T^* \in E_2(X, .)$; if not E adds (L^*, R, X, S^*, T^*) to the set $\mathcal{F}$ of failed attempts and tries again. On encountering the first success E discards the successful tuple and returns to $\mathcal{A}$ the honest tuple (L, R, X, S, T) (where $L = E_1^{-1}(R, X)$ and $(S, T) = \Psi(L, R)$). If it remains unsuccessful after all n attempts, it directly returns the honest tuple to $\mathcal{A}$ and stops.

3.3.2 Bounding $\Delta_{\mathcal{A}}\left[\mathcal{G}_0, \mathcal{G}_{1/2}\right]$

Recall that in $\mathcal{G}_0$ the adversary $\mathcal{A}$ interacts with $(\pi, \mathscr{S}^{\pi})$ and in $\mathcal{G}_{1/2}$ it interacts with $(\widetilde{\pi}, \mathscr{S}^{\widetilde{\pi}})$. Suppose $\mathcal{A}$ makes q queries in total (it doesn't make any repeated or redundant queries). According to our condition, $\mathcal{A}$ gives the construction queries to $\mathscr{S}$ after completing all queries. So there are in total exactly q 5-tuples (L, R, X, S, T) in the transcript. For determining each of these q 5-tuples, exactly one of π and π^{-1} is invoked in $\mathcal{G}_0$, and similarly exactly one of $\widetilde{\pi}$

and $\widetilde{\pi}^{-1}$ is invoked in $\mathcal{G}_{1/2}$. Let us assume the construction has been queried q_f^* times in the forward direction and q_b^* times in the backward direction (when asked directly by the adversary, or through simulator queries) where $q^* = q_f^* + q_b^* \leq nq$. Let τ be the final adversary transcript.

The Bad Event. In the game $\mathcal{G}_{1/2}$, we say the event bad^* has occurred if at any point during the query phase $\widetilde{\pi}$ returns $\perp$. This can be in response to a query from $\mathcal{A}$ or a query from $\mathscr{S}$. We call τ good if it can be obtained in $\mathcal{G}_{1/2}$ without encountering the event bad^*. In the next lemma, we derive an upper bound on $\Pr[\mathsf{bad}^*]$.

Lemma 2.

$$\Pr[\mathsf{bad}^*] \leq \frac{n^2q^2}{2^{2n}}.$$

Proof. There are three cases which can lead to $\widetilde{\pi}$ returning $\perp$; we go over them one by one and bound their probabilities.

Case 1: For a forward query (L, R), $\widetilde{\pi}$ picks an (S, T) such that

$$(S, T) = \widetilde{\pi}(L', R')$$

for an earlier forward query (L', R'). Call the probability of this p_1. There are q_f^* forward queries. Any pair of them can provide the required collision. So

$$p_1 \leq \frac{q_f^{*2}}{2^{2n+1}}. \tag{3.1}$$

Case 2: For a backward query (S, T), $\widetilde{\pi}$ picks an (L, R) such that

$$(L, R) = \widetilde{\pi}^{-1}(S', T')$$

for an earlier backward query (S', T'). Call the probability of this p_2. There are q_b^* backward queries. Any pair of them can provide the required collision. So

$$p_2 \leq \frac{q_b^{*2}}{2^{2n+1}}. \tag{3.2}$$

Case 3: For a forward query (L, R), $\widetilde{\pi}$ picks an (S, T) which was itself an earlier backward query; or for a backward query (S, T), $\widetilde{\pi}$ picks an (L, R) which was itself an earlier backward query. Call the probability of this p_3. There are q_f^* forward queries and q_b^* backward queries. Hence there are at most $q_f^* q_b^*$ possible collision pairs, giving

$$p_3 \leq \frac{q_f^* q_b^*}{2^{2n}}. \tag{3.3}$$

By union-bound,

$$\Pr[\mathsf{bad}^*] \leq \sum_{i=1}^{3} p_i. \tag{3.4}$$

Substituting Eqs. 3.1, 3.2 and 3.3 in Eq. 3.4 gives us

$$\Pr[\mathsf{bad}^*] \leq \frac{q_f^{*2}}{2^{2n+1}} + \frac{q_b^{*2}}{2^{2n+1}} + \frac{q_f^* q_b^*}{2^{2n}} = \frac{q^{*2}}{2^{2n+1}} \leq \frac{n^2q^2}{2^{2n+1}},$$

which is the bound claimed in the lemma.

□

Probabilities of a Good Transcript. The next task is to bound the ratio of the probabilities of a good transcript in $\mathcal{G}_0$ and $\mathcal{G}_{1/2}$. This we do in the following lemma.

Lemma 3. *For any good transcript τ,*

$$\Pr\left[\mathcal{G}_0 \textit{ yields } \tau\right] \geq \Pr\left[\mathcal{G}_{1/2} \textit{ yields } \tau\right].$$

Proof. We fix a good transcript τ. For $b \in \{0,1\}$, the event $\{\mathcal{G}_b \text{ yields } \tau\}$ has two independent sources of randomness: the random coin of the oracle (π or $\widetilde{\pi}$) when called by $\mathcal{A}$ or the simulator, and the internal random coin of the simulator $\mathscr{S}$. Accordingly, τ can be seen as consisting of two parts: τ_p, which is dependent on the random coin of the oracle; and τ_s, which is dependent on the random coin of the simulator. Then we can say that

$$\Pr\left[\mathcal{G}_0 \text{ yields } \tau\right] = \Pr\left[\pi \text{ yields } \tau_p\right] \cdot \Pr\left[\mathscr{S} \text{ yields } \tau_s\right],$$

$$\Pr\left[\mathcal{G}_{1/2} \text{ yields } \tau\right] = \Pr\left[\widetilde{\pi} \text{ yields } \tau_p\right] \cdot \Pr\left[\mathscr{S} \text{ yields } \tau_s\right].$$

To complete the proof, we need to show that

$$\Pr\left[\pi \text{ yields } \tau_p\right] \geq \Pr\left[\widetilde{\pi} \text{ yields } \tau_p\right]. \tag{3.5}$$

In $\mathcal{G}_0$, in the process of yielding τ_p, π is called exactly q times, including forward and backward queries by $\mathcal{A}$ and $\mathscr{S}$. If the i-th query is an (L,R) query, the response (S,T) is chosen distinct from all previous (S,T) values, which are themselves all distinct since τ is a good transcript. Similarly, if the i-th query is an (S,T) query, the response (L,R) is chosen distinct from all previous (L,R) values, which are again themselves all distinct. Thus, conditioned on the first $(i-1)$ responses, the probability of the i-th response of π matching that of τ_p is always $1/(2^{2n} - i + 1)$. Recalling that there are q^* distinct queries to π in τ, we have

$$\Pr\left[\pi \text{ yields } \tau_p\right] = \frac{1}{2^{2n} \cdot (2^{2n}-1) \cdot \ldots \cdot (2^{2n} - q^* + 1)} \geq \frac{1}{(2^{2n})^{q^*}}. \tag{3.6}$$

Similarly, in $\mathcal{G}_{1/2}$, in the process of yielding τ_p, $\widetilde{\pi}$ is called exactly q times. Here, irrespective of the direction of the query, the responses are chosen uniformly from $\{0,1\}^{2n}$ and independently of each other. Thus,

$$\Pr\left[\widetilde{\pi} \text{ yields } \tau_p\right] = \frac{1}{(2^{2n})^{q^*}}. \tag{3.7}$$

The right hand side of Eq. 3.7 is the right hand side of Eq. 3.6, thus establishing Eq. 3.5 and completing the proof of the lemma. □

Finally, we find a bound on $\Delta_{\mathcal{A}}\left[\mathcal{G}_0, \mathcal{G}_{1/2}\right]$ in the following lemma, using the results derived above.

Lemma 4.

$$\Delta_{\mathcal{A}}\left[\mathcal{G}_0, \mathcal{G}_{1/2}\right] \leq \frac{n^2 q^2}{2^{2n+1}}.$$

Proof. The result follows from Theorem 1, after substituting $\epsilon_1 = n^2q^2/2^{2n+1}$ (from Lemma 2) and $\epsilon_2 = 0$ (from Lemma 3). □

3.4 Bounding $\Delta_{\mathcal{A}}\left[\mathcal{G}_{1/2}, \mathcal{G}_1\right]$ and Deriving the Final Bound

Since the analysis of this section is the most involved part of this work, we begin by introducing some notations to describe certain sets and counters which will often be referred to later in the section. In the description that follows, we abuse the term simulator queries to mean queries to $\mathscr{S}$ when talking about $\mathcal{G}_{1/2}$ and queries to E when talking about $\mathcal{G}_1$; when we specifically talk about one of the two we explicitly name $\mathscr{S}$ or E. Similarly, we say construction queries to mean queries (in either direction) to $\widetilde{\pi}$ when talking about $\mathcal{G}_{1/2}$ and queries to Ψ when talking about $\mathcal{G}_1$. Unless otherwise mentioned, when we refer to the construction queries up to a certain point, we include the failed queries, if any.

We will use the indicator set $I_{A,B}$ to denote the indices of all $\{A, B\}$ simulator queries. As in the previous proof, let q_f^* denote all forward construction queries, and q_b^* denote all backward construction queries, with $q^* = q_f^* + q_b^* \leq nq$. Let τ_{con}^i denote the set of 4-tuples (L, R, S, T) that were revealed to $\mathcal{A}$ through construction queries *before* the i-th simulator query. For $i \in I_{R,X}$, if the i-th simulator query is $\{R_0, X_0\}$, we define the following:

$$\begin{aligned}
\mathcal{B}_{R_0,X_0}^{0,i} &:= E_1^{-1}(R_0, \cdot),\\
\mathcal{B}_{R_0,X_0}^{1,i} &:= \left\{L \mid \text{some } (L, R_0, S, T) \in \tau_{\mathsf{con}}^i, T \in E_2(X_0, \cdot)\right\},\\
\mathcal{B}_{R_0,X_0}^{2,i} &:= \left\{L \mid \text{some } (L, R_0, S, T) \in \tau_{\mathsf{con}}^i, T \notin E_2(X_0, \cdot)\right\},\\
\mathcal{B}_{R_0,X_0}^{3,i} &:= \{0,1\}^n \setminus \bigcup_{j=0}^{2} \mathcal{B}_{R_0,X_0}^{j,i},\\
\mu_{R_0,X_0}^{j,i} &:= \left|\mathcal{B}_{R_0,X_0}^{j,i}\right| \text{ for } 0 \leq j \leq 3.
\end{aligned}$$

Similarly, for $i \in I_{X,T}$, if the i-th simulator query is $\{X_0, T_0\}$, we define the following:

$$\begin{aligned}
\mathcal{B}_{X_0,T_0}^{0,i} &:= E_3(T_0, \cdot),\\
\mathcal{B}_{X_0,T_0}^{1,i} &:= \left\{S \mid \text{some } (L, R, S, T_0) \in \tau_{\mathsf{con}}^i, R \in E_2^{-1}(X_0, \cdot)\right\},\\
\mathcal{B}_{X_0,T_0}^{2,i} &:= \left\{S \mid \text{some } (L, R, S, T_0) \in \tau_{\mathsf{con}}^i, R \notin E_2^{-1}(X_0, \cdot)\right\},\\
\mathcal{B}_{X_0,T_0}^{3,i} &:= \{0,1\}^n \setminus \bigcup_{j=0}^{2} \mathcal{B}_{X_0,T_0}^{j,i},\\
\mu_{X_0,T_0}^{j,i} &:= \left|\mathcal{B}_{X_0,T_0}^{j,i}\right| \text{ for } 0 \leq j \leq 3.
\end{aligned}$$

Next, we describe some notation specific to one of the two games $\mathcal{G}_{1/2}$ and $\mathcal{G}_1$. In the query phase of $\mathcal{G}_{1/2}$, let $\tau_{\mathscr{S}}^i$ denote the set of 5-tuples (L, R, X, S, T) that were revealed to $\mathcal{A}$ through $\mathscr{S}$ queries *before* the i-th query to $\mathscr{S}$. Let $(L_0, R_0, X_0, S_0, T_0)$ be the 5-tuple revealed on the i-th simulator query, irrespective of the type of this query. We define the following counters:

$$\begin{aligned}
r_i &:= \left|\left\{(L, X, S, T) \mid (L, R_0, X, S, T) \in \tau_{\mathscr{S}}^i\right\}\right|,\\
t_i &:= \left|\left\{(L, R, X, S) \mid (L, R, X, S, T_0) \in \tau_{\mathscr{S}}^i\right\}\right|,\\
x_i &:= \left|\left\{(L, R, S, T) \mid (L, R, X_0, S, T) \in \tau_{\mathscr{S}}^i\right\}\right|,\\
m_i &:= \min(r_i, t_i),
\end{aligned}$$

$$M_i := \max(r_i, t_i).$$

Next, in the query phase of $\mathcal{G}_1$, let $\boldsymbol{\tau}_E^i$ denote the 5-tuples that were sampled *before* the i-th query. (We note that in $\mathcal{G}_1$, for every query, a distinct 5-tuple (L, R, X, S, T) is sampled, irrespective of the nature and direction of the query; when E is queried, the entire 5-tuple is immediately revealed to $\mathcal{A}$, and when Ψ is queried, only (L, R, S, T) is revealed immediately, and X is revealed at the end of the query phase; but in both cases, the sampling is done on-the-fly.) Suppose $(L_0, R_0, X_0, S_0, T_0)$ is the 5-tuple sampled on the i-th query. We define the following counters:

$$\begin{aligned}
r_i' &:= \Big| \big\{ (L, X, S, T) \mid (L, R_0, X, S, T) \in \boldsymbol{\tau}_E^i \big\} \Big|, \\
t_i' &:= \Big| \big\{ (L, R, X, S) \mid (L, R, X, S, T_0) \in \boldsymbol{\tau}_E^i \big\} \Big|, \\
x_i' &:= \Big| \big\{ (L, R, S, T) \mid (L, R, X_0, S, T) \in \boldsymbol{\tau}_E^i \big\} \Big|.
\end{aligned}$$

Finally, let $\kappa_{\longrightarrow}^{\mathsf{mcoll}}$ denote the maximum number of multi-collisions on T over the forward construction queries, and $\kappa_{\longleftarrow}^{\mathsf{mcoll}}$ denote the maximum number of multi-collisions on R over the backward construction queries.

3.4.1 Bad Events

We define the following bad events for $\mathcal{G}_{1/2}$:

bad_1: $\sum_{i=1}^{q^*} m_i \geq q$;

bad_2: $\sum_{i=1}^{q^*} m_i < q$, $\mathscr{S}$ never returns $\perp$, and at some query in $I_{L,R} \bigcup I_{S,T}$, $\mathscr{S}$ violates the tweakable random permutation property, i.e it reveals an $X \in E_1(R, \cdot) \bigcup E_3^{-1}(T, \cdot)$;

bad_3: $\mathscr{S}$ returns $\perp$ at some point;

bad_4: $\max\left(\kappa_{\longrightarrow}^{\mathsf{mcoll}}, \mathcal{B}\right) \geq n$;

bad_5: $\widetilde{\boldsymbol{\pi}}$ returns $\perp$ at some point.

Finally, we define

$$\mathsf{bad} := \bigcup_{i=1}^{5} \mathsf{bad}_i.$$

Thus, by union-bound,

$$\Pr\left[\mathsf{bad}\right] \leq \sum_{i=1}^{5} \Pr\left[\mathsf{bad}_i\right]. \tag{3.8}$$

As before, we call $\boldsymbol{\tau}$ good if it can be obtained in $\mathcal{G}_{1/2}$ without encountering the event bad.

3.4.2 Probability Bound for Bad Events

In the next lemma, we derive an upper bound for $\Pr[\mathsf{bad}]$.

Lemma 5. *For $n \geq 6$,*

$$\Pr[\mathsf{bad}] \leq \frac{n^2 q}{2^n} + \frac{n^2 q^2}{2^{2n+1}}.$$

Proof. We bound the probabilities of the bad events $\mathsf{bad}_1, \ldots, \mathsf{bad}_5$ separately.

bad_1: We define the indicator random variables $Y_{i,j}$ and $Z_{i,j}$ as follows :

$$Y_{i,j} := \begin{cases} 1, & \text{if } R_i = R_j \\ 0, & \text{otherwise} \end{cases}$$

and

$$Z_{i,j} := \begin{cases} 1, & \text{if } T_i = T_j \\ 0, & \text{otherwise} \end{cases}$$

Note that the adversary can fix only one of R and T in any query while the other is chosen uniformly at random. So we have either $\Pr[Y_{i,j} = 1] = 1/2^n$ for all $j < i$, or $\Pr[Z_{i,j} = 1] = 1/2^n$ for all $j < i$. Note that

$$r_i = \sum_{j<i} Y_{i,j}, \quad t_i = \sum_{j<i} Z_{i,j}.$$

Now let $\mathbb{E}[Y]$ denote the expectation of a random variable Y. Then for all i, either

$$\mathbb{E}[r_i] = \mathbb{E} \sum_{j<i} [Y_{i,j}] = \sum_{j<i} \mathbb{E}[Y_{i,j}] = \sum_{j<i} \frac{1}{2^n} = \frac{i-1}{2^n},$$

or $\mathbb{E}(t_i) = (i-1)/2^n$ (by similar reasoning). Now,

$$\begin{aligned} \mathbb{E}\left[\sum_{1}^{q^*} m_i\right] &= \sum_{1}^{q^*} \mathbb{E}\left[\min(r_i, t_i)\right] \\ &\leq \sum_{1}^{q^*} \min\left(\mathbb{E}[r_i], \mathbb{E}[t_i]\right) \leq \sum_{1}^{q^*} \frac{i-1}{2^n} = \frac{q^*(q^*-1)}{2^{n+1}}. \end{aligned}$$

Now, as $q^* \leq nq$ from Markov's inequality, we have

$$\begin{aligned} \Pr\left[\mathsf{bad}_1\right] &= \Pr\left[\sum_{1}^{q^*} m_i \geq q\right] \\ &\leq \frac{1}{q} \cdot \mathbb{E}\left[\sum_{1}^{q^*} m_i\right] \leq \frac{1}{q} \cdot \frac{nq(nq-1)}{2^{n+1}} \leq \frac{n^2 q}{2^{n+1}}. \end{aligned} \tag{3.9}$$

bad_2: We recall that for an $\{L, R\}$ or $\{S, T\}$ query, $\mathscr{S}$ picks X from $\{0,1\}^n \backslash \max(E_1(R,\cdot), E_3^{-1}(T,\cdot))$. Thus, bad_2 can happen only when X is picked from the smaller of the two sets $E_1(R,\cdot)$ and $E_3^{-1}(T,\cdot)$. In the i-th query the probability of this event is bounded by $m_i/(2^n - M_i)$. So by applying union-bound, and the fact that $q \leq 2^n - 1$, we have,

$$\Pr[\mathsf{bad}_2] \leq \sum_{1}^{q^*} \frac{m_i}{2^n - M_i} \leq \sum_{1}^{q^*} \frac{m_i}{2^n - q} \leq \frac{q}{2^n - q} \leq \frac{2q}{2^n}. \tag{3.10}$$

bad_3: Consider a particular (R, X) query (R_0, X_0). The probability that $\mathscr{S}$ returns $\perp$ is at most $(\upmu^{2,i}_{R_0,X_0} + n + \upmu^{3,i}_{R_0,X_0} \cdot (x_i/2^n))^n/(2^n)^n$, since the number of sampling options which guarantee failure starts at $\upmu^{2,i}_{R_0,X_0}$ and can go up to $\upmu^{2,i}_{R_0,X_0} + n$ (it increases by one each time a new L sampled from $\upmu^{3,i}_{R_0,X_0}$ leads to failure, which can happen at most n times), and in addition, there are $\upmu^{3,i}_{R_0,X_0}$ options which lead to failure with probability $x_i/2^n$. Now

$$\begin{aligned}
&\frac{(\upmu^{2,i}_{R_0,X_0} + n + \upmu^{3,i}_{R_0,X_0} \cdot (nx_i/2^n))^n}{(2^n)^n} \\
&\le \frac{(nx_i + n + \upmu^{3,i}_{R_0,X_0} \cdot (nx_i/2^n))^n}{(2^n)^n} \\
&\le \frac{(nq + n + 2^n \cdot (nq/2^n))^n}{(2^n)^n} \\
&\le \left(\frac{1}{n} + \frac{n}{2^n} + \frac{1}{n}\right)^n \le \frac{1}{2^n},
\end{aligned}$$

the last two inequalities following from the conditions $q \le 2^n/n^2$ and $n \ge 6$ respectively. The analysis for (X, T) queries will be similar. As there are at most q many such queries, by using union-bound we have

$$\Pr[\mathsf{bad}_3] \le \frac{q}{2^n}. \tag{3.11}$$

bad_4: For any forward query (L, R) and any T we have

$$\Pr[\widetilde{\pi}(L, R) \in \{0,1\}^n \times T] = \frac{1}{2^n}.$$

By applying union-bound and then summing over all such T we have

$$\begin{aligned}
\Pr\left[\kappa^{\mathsf{mcoll}}_{\longrightarrow} \ge n\right] &\le \sum_{T \in \{0,1\}^n} \frac{1}{(2^n)^n} \cdot \binom{q^*}{n} \\
&\le 2^n \cdot \frac{q^{*n}}{(2^n)^n} \\
&\le \frac{(nq)^n}{2^{n(n-1)}} \le \frac{q}{2^n},
\end{aligned}$$

where the last line of the inequality holds for $n \ge 4$, as $q \le 2^n/n$.

Similarly, we can show that

$$\Pr\left[\mathcal{B} \ge n\right] \le \frac{q}{2^n}.$$

Then we have

$$\begin{aligned}
\Pr[\mathsf{bad}_4] &= \Pr\left[\max\left(\kappa^{\mathsf{mcoll}}_{\longrightarrow}, \mathcal{B}\right) \ge n\right] \\
&\le \Pr\left[\kappa^{\mathsf{mcoll}}_{\longrightarrow} \ge n\right] + \Pr\left[\mathcal{B} \ge n\right] \le \frac{2q}{2^n}.
\end{aligned} \tag{3.12}$$

bad_5: This case is identical to bad^* in Lemma 2 (see p. 25), and recalling the bound obtained there we have

$$\Pr[\mathsf{bad}_5] = \Pr\left[\mathsf{bad}^*\right] \le \frac{n^2q^2}{2^{2n+1}}. \tag{3.13}$$

Substituting the bounds from Eqs. 3.9 through 3.13 in Eq. 3.8 gives us

$$\begin{aligned}\Pr[\mathsf{bad}] &\le \frac{n^2 q}{2^{n+1}} + \frac{5q}{2^n} + \frac{n^2 q^2}{2^{2n+1}} \\ &= \left(\frac{n^2}{2} + 5\right) \cdot \frac{q}{2^n} + \frac{n^2 q^2}{2^{2n+1}} \le \frac{n^2 q}{2^n} + \frac{n^2 q^2}{2^{2n+1}},\end{aligned}$$

which completes the proof of Lemma 5. □

3.4.3 Probabilities of a Good Transcript

In the final part of our analysis, we derive a bound for the ratio of probabilities of a good transcript in $\mathcal{G}_1$ and $\mathcal{G}_{1/2}$.

Lemma 6. *For any good transcript* $\boldsymbol{\tau}$*,*

$$\Pr\left[\mathcal{G}_1 \text{ yields } \boldsymbol{\tau}\right] \ge \left(1 - \frac{6nq}{2^n}\right) \cdot \Pr\left[\mathcal{G}_{1/2} \text{ yields } \boldsymbol{\tau}\right].$$

Proof. We first turn our attention to bounding $\Pr\left[\mathcal{G}_{1/2} \text{ yields } \boldsymbol{\tau}\right]$. In $\mathcal{G}_{1/2}$ there are two independent sources of randomness: the random coin of $\widetilde{\boldsymbol{\pi}}$, and the internal random coin of $\mathscr{S}$. Since bad is not encountered, all responses of $\widetilde{\boldsymbol{\pi}}$ are independent and uniformly random. For calculating the probability of $\mathscr{S}$ outputting on its i-th query the 5-tuple $(L_0, R_0, X_0, S_0, T_0)$ (as well as the failure set $\mathcal{F}_0$ of size k^i when $i \in I_{R,X} \cup I_{X,T}$) that matches the corresponding one in $\boldsymbol{\tau}$, we condition on the transcript revealed to $\mathcal{A}$ over the first $i-1$ queries to $\mathscr{S}$ and all direct queries to $\widetilde{\boldsymbol{\pi}}$ up to that point; call this entire transcript-so-far $\boldsymbol{\tau}^i$, and let p^i denote this conditional probability of $\mathscr{S}$ revealing $(L_0, R_0, X_0, S_0, T_0)$ and $\mathcal{F}_0$ (if applicable) at the i-th query. Then

$$\Pr\left[\mathcal{G}_{1/2} \text{ yields } \boldsymbol{\tau}\right] = \prod_{i=1}^{q} p^i. \tag{3.14}$$

We consider each type of query separately:

$i \in I_{L,R}$: We recall that here $\mathscr{S}$ proceeds in two steps: it first sets $(S,T) = \widetilde{\boldsymbol{\pi}}(L_0, R_0)$, and then samples X uniformly at random from the set $\{0,1\}^n \setminus \max(E_1(R_0,\cdot), E_3^{-1}(T_0,\cdot))$, which is of size $2^n - M_i$. Thus,

$$\begin{aligned}p^i &= \Pr\left[(X,S,T) = (X_0,S_0,T_0) \;\middle|\; \boldsymbol{\tau}^i\right] \\ &= \Pr\left[X = X_0 \;\middle|\; \boldsymbol{\tau}^i\right] \cdot \Pr\left[(S,T) = (S_0,T_0) \;\middle|\; \boldsymbol{\tau}^i\right]\end{aligned} \tag{3.15}$$

by the independence of the random coins of $\widetilde{\boldsymbol{\pi}}$ and $\mathscr{S}$. We know that

$$\Pr\left[X = X_0 \;\middle|\; \boldsymbol{\tau}^i\right] = \frac{1}{2^n - M_i}. \tag{3.16}$$

Since $\mathscr{S}$ has just been queried with (L_0, R_0), we know that (L_0, R_0) does not occur as part of any other 5-tuple fixed by $\mathscr{S}$. Thus, even if $(L_0, R_0, S_0, T_0) \in \boldsymbol{\tau}^i$, it does not affect any other query to $\mathscr{S}$. So we can assume for simplicity that $(L_0, R_0, S_0, T_0) \notin \boldsymbol{\tau}^i$. Thus,

$$\Pr\left[(S,T) = (S_0,T_0) \;\middle|\; \boldsymbol{\tau}^i\right] = \Pr\left[(S,T) = (S_0,T_0)\right] = \frac{1}{2^{2n}}. \tag{3.17}$$

(Without the simplifying assumption, for the case of $(L_0, R_0, S_0, T_0) \in \boldsymbol{\tau}^i$, this conditional probability would instead be 1, but the $1/2^{2n}$ would be obtained from the probability of that previous $\widetilde{\boldsymbol{\pi}}$ query, so this does not affect the overall calculations.) Combining Eqs. 3.15, 3.16 and 3.17, we get

$$p^i = \frac{1}{2^{2n} \cdot (2^n - M_i)}. \tag{3.18}$$

$i \in I_{S,T}$: The analysis here goes just as in the case of $i \in I_{L,R}$, so we obtain

$$p^i = \frac{1}{2^{2n} \cdot (2^n - M_i)}. \tag{3.19}$$

$i \in I_{R,X}$: Here, $\mathscr{S}$ first samples L uniformly at random from the set $\{0,1\}^n \backslash E_1^{-1}(R_0, \cdot))$, which is of size $2^n - r_i$; then it provisionally sets $(S, T) = \widetilde{\boldsymbol{\pi}}(L, R_0)$, and halts if $T \notin E_2(X_0, \cdot)$; else it adds (L, R_0, S, T) to the failure set $\mathcal{F}$, resamples L, and repeats the steps. Since $\boldsymbol{\tau}$ is good, we know $\mathscr{S}$ halts at some point; let L^* be the value of L for which it halts. Now there are two cases: $L_0 \in \mathcal{B}^{2,i}_{R_0,X_0}$, and $L_0 \in \mathcal{B}^{3,i}_{R_0,X_0}$. Thus, we can write

$$\begin{aligned} p^i &= \Pr\left[(L^*, S, T) = (L_0, S_0, T_0), \mathcal{F} = \mathcal{F}_0 \;\middle|\; \boldsymbol{\tau}^i\right] \\ &= \Pr\left[(S, T) = (S_0, T_0) \;\middle|\; L^* = L_0, \mathcal{F} = \mathcal{F}_0, \boldsymbol{\tau}^i\right] \cdot \Pr\left[L^* = L_0, \mathcal{F} = \mathcal{F}_0 \;\middle|\; \boldsymbol{\tau}^i\right]. \end{aligned} \tag{3.20}$$

As in the previous case, we have

$$\Pr\left[(S, T) = (S_0, T_0) \;\middle|\; L^* = L_0, \mathcal{F} = \mathcal{F}_0, \boldsymbol{\tau}^i\right] = \Pr\left[(S, T) = (S_0, T_0)\right] = \frac{1}{2^{2n}}. \tag{3.21}$$

We observe that

$$\begin{aligned} \Pr\left[L^* = L_0, \mathcal{F} = \mathcal{F}_0 \;\middle|\; \boldsymbol{\tau}^i\right] &= \Pr\left[L^* = L_0 \;\middle|\; \mathcal{F} = \mathcal{F}_0, \boldsymbol{\tau}^i\right] \cdot \Pr\left[\mathcal{F} = \mathcal{F}_0 \;\middle|\; \boldsymbol{\tau}^i\right] \\ &= \frac{1}{2^n - r^i} \cdot \left(\frac{1}{2^n - r^i}\right)^{k^i} = \frac{1}{(2^n - r^i)^{k^i + 1}}. \end{aligned} \tag{3.22}$$

From Eqs. 3.20, 3.21 and 3.22 we get

$$p^i \leq \frac{1}{2^{2n} \cdot (2^n - r^i)^{k^i + 1}}. \tag{3.23}$$

$i \in I_{X,T}$: The analysis here goes just as in the case of $i \in I_{R,X}$, so we obtain

$$p^i \leq \frac{1}{2^{2n} \cdot (2^n - t^i)^{k^i + 1}}. \tag{3.24}$$

Substituting Eqs. 3.18, 3.19, 3.23 and 3.24 in Eq. 3.14 gives us

$$\begin{aligned} \Pr\left[\mathcal{G}_{1/2} \text{ yields } \boldsymbol{\tau}\right] \leq & \prod_{i \in I_{L,R} \cup I_{S,T}} \frac{1}{2^{2n} \cdot (2^n - M_i)} \\ & \cdot \prod_{i \in I_{R,X}} \frac{1}{2^{2n} \cdot (2^n - r^i)^{k^i + 1}} \cdot \prod_{i \in I_{X,T}} \frac{1}{2^{2n} \cdot (2^n - t^i)^{k^i + 1}}. \end{aligned} \tag{3.25}$$

Bounding $\Pr[\mathcal{G}_1$ **yields** $\boldsymbol{\tau}]$**.** We continue using the definition of $\boldsymbol{\tau}^i$ from the analysis of $\mathcal{G}_{1/2}$, and let p'^i denote the conditional probability of E revealing $(L_0, R_0, X_0, S_0, T_0)$ as well as $\mathcal{F}_0$ (when $i \in I_{R,X} \cup I_{X,T}$) at the i-th query given $\boldsymbol{\tau}^i$. Then

$$\Pr[\mathcal{G}_1 \text{ yields } \boldsymbol{\tau}] = \prod_{i=1}^{q} p'^i. \tag{3.26}$$

For $i \in I_{L,R} \cup I_{S,T}$, E reveals $(L_0, R_0, X_0, S_0, T_0)$ if and only if $E_1(R_0, L_0) = X_0$, $E_2(X_0, R_0) = T_0$, $E_3(T_0, X_0) = S_0$. Thus,

$$p'^i = \frac{1}{(2^n - r'_i) \cdot (2^n - x'_i) \cdot (2^n - t'_i)}. \tag{3.27}$$

For $i \in I_{R,X} \cup I_{X,T}$, in order for E to reveal $(L_0, R_0, X_0, S_0, T_0)$ and $\mathcal{F}_0$ (of size k^i), the $(k^i + 1)$-th attempt of E needs to succeed; call the probability of this p'^i_s. For $i \in I_{R,X}$, calculating the probability of $\mathcal{F}_0$ as in the analysis of $\mathcal{G}_{1/2}$ and that of $(L_0, R_0, X_0, S_0, T_0)$ as in Eq. 3.27, we have

$$p'^i = \frac{1}{(2^n - r'_i) \cdot (2^n - x'_i) \cdot (2^n - t'_i) \cdot (2^n - t_i)^{k^i}} \cdot p'^i_s. \tag{3.28}$$

Let L^* be the L sampled in the successful attempt, and let $p'^i_s(L^*)$ denote the success probability conditioned on L^*. For $L^* \in \mathcal{B}^{1,i}_{X_0,T_0}$, $p'^i_s(L^*) = 0$ and for $L^* \in \mathcal{B}^{2,i}_{X_0,T_0}$, $p'^i_s(L^*) = 1$. When $L^* \in \mathcal{B}^{3,i}_{X_0,T_0}$, an X is first sampled from $\{0,1\}^n \setminus E_1(R_0, \cdot)$, then a T is sampled from $\{0,1\}^n \setminus E_2(X, \cdot)$; it is a success if $T \notin E_2(X_0, \cdot)$. Define

$$[X]_i := \Big| \big\{ (L, R, S, T) \bigm| (L, R, X, S, T) \in \boldsymbol{\tau}^i_{\mathscr{S}} \big\} \Big|.$$

Then the probability of success given X is $(2^n - x_i - [X]_i)/(2^n - [X]_i)$. Taking expectation over X gives us

$$\begin{aligned}
p'^i_s(L^*) &= \frac{1}{2^n - r_i} \cdot \sum_{X \notin E_1(R_0, \cdot)} \frac{2^n - x_i - [X]_i}{2^n - [X]_i} \\
&\geq \frac{1}{2^n - r_i} \cdot \sum_{X \notin E_1(R_0, \cdot)} \frac{2^n - x_i - [X]_i}{2^n} \\
&= \frac{1}{2^n - r_i} \cdot \sum_{X \notin E_1(R_0, \cdot)} \frac{2^n - x_i}{2^n} - \sum_{X \notin E_1(R_0, \cdot)} \frac{[X]_i}{2^n \cdot (2^n - r_i)} \\
&\geq \frac{2^n - x_i}{2^n} - \frac{nq}{2^n \cdot (2^n - r_i)} \geq \frac{2^n - x_i - 1}{2^n} = 1 - \frac{x_i + 1}{2^n}.
\end{aligned} \tag{3.29}$$

Thus, $p'^i_s(L^*) \geq 1 - (x_i + 1)/2^n$ for all choices of L^* except when $L^* \in \mathcal{B}^{1,i}_{X_0,T_0}$. Taking expectation over L^* gives us

$$\begin{aligned}
p'^i_s &\geq \frac{1}{2^n - r_i} \cdot \sum_{L^* \in E^{-1}(R_0, \cdot) \setminus \mathcal{B}^{1,i}_{X_0,T_0}} \left(1 - \frac{x_i + 1}{2^n}\right) \\
&= \frac{1}{2^n - r_i} \cdot \left(2^n - r_i - \mu^{1,i}_{X_0,T_0}\right) \cdot \left(1 - \frac{x_i + 1}{2^n}\right) \\
&= \left(1 - \frac{\mu^{1,i}_{X_0,T_0}}{2^n - r_i}\right) \cdot \left(1 - \frac{x_i + 1}{2^n}\right).
\end{aligned} \tag{3.30}$$

From Eqs. 3.28 and 3.30 we have

$$p'^{i} \geq \frac{1}{(2^n - r'_i)\cdot(2^n - x'_i)\cdot(2^n - t'_i)\cdot(2^n - r_i)^{k^i}} \cdot \left(1 - \frac{\mu^{1,i}_{X_0,T_0}}{2^n - r_i}\right) \cdot \left(1 - \frac{x_i + 1}{2^n}\right). \tag{3.31}$$

Similarly for $i \in I_{X,T}$ we have

$$p'^{i} \geq \frac{1}{(2^n - r'_i)\cdot(2^n - x'_i)\cdot(2^n - t'_i)\cdot(2^n - t_i)^{k^i}} \cdot \left(1 - \frac{\mu^{1,i}_{R_0,X_0}}{2^n - t_i}\right) \cdot \left(1 - \frac{x_i + 1}{2^n}\right). \tag{3.32}$$

Substituting Eqs. 3.27, 3.31 and 3.32 in Eq. 3.26 gives us

$$\begin{aligned}\Pr\left[\mathcal{G}_1 \text{ yields } \tau\right] \geq & \prod_{i \in I_{L,R} \cup I_{S,T} \cup I_{R,X} \cup I_{X,T}} \frac{1}{(2^n - r'_i)\cdot(2^n - x'_i)\cdot(2^n - t'_i)} \\ & \cdot \prod_{i \in I_{R,X}} \frac{1}{\cdot\,(2^n - r_i)^{k^i}} \cdot \left(1 - \frac{\mu^{1,i}_{X_0,T_0}}{2^n - r_i}\right) \cdot \left(1 - \frac{x_i + 1}{2^n}\right) \\ & \cdot \prod_{i \in I_{X,T}} \frac{1}{\cdot\,(2^n - t_i)^{k^i}} \cdot \left(1 - \frac{\mu^{1,i}_{R_0,X_0}}{2^n - t_i}\right) \cdot \left(1 - \frac{x_i + 1}{2^n}\right).\end{aligned} \tag{3.33}$$

Ratio of Probabilities. Finally, we look at the ratio

$$\rho := \frac{\Pr\left[\mathcal{G}_1 \text{ yields } \tau\right]}{\Pr\left[\mathcal{G}_{1/2} \text{ yields } \tau\right]}. \tag{3.34}$$

We first observe that $\{r'_i\}, \{x'_i\}, \{t'_i\}$ are just a re-ordering of $\{r_i\}, \{x_i\}, \{t_i\}$, so

$$\prod_i \frac{1}{(2^n - r'_i)\cdot(2^n - x'_i)\cdot(2^n - t'_i)} = \prod_i \frac{1}{(2^n - r_i)\cdot(2^n - x_i)\cdot(2^n - t_i)}, \tag{3.35}$$

where both products are taken over $I_{L,R} \cup I_{S,T} \cup I_{R,X} \cup I_{X,T}$. We observe further that for any i,

$$2^n \cdot \left(2^n - M^i\right) \geq \left(2^n - r^i\right) \cdot \left(2^n - t^i\right). \tag{3.36}$$

Finally, we observe that

$$\left(1 - \frac{x_i + 1}{2^n}\right) \cdot \frac{2^n}{2^n - x^i} = 1 - \frac{1}{2^n - x^i}. \tag{3.37}$$

After substituting Eqs. 3.25 and. 3.26 in Eq. 3.34, applying the observations in Eqs. 3.35, 3.36 and 3.37, and simplifying a little, we arrive at the inequality

$$\rho \geq \prod_{i \in I_{R,X}} \left(1 - \frac{\mu^{1,i}_{X_0,T_0}}{2^n - r_i}\right) \cdot \prod_{i \in I_{X,T}} \left(1 - \frac{\mu^{1,i}_{R_0,X_0}}{2^n - t_i}\right) \cdot \prod_{i \in I_{R,X} \cup I_{X,T}} \left(1 - \frac{1}{2^n - x^i}\right). \tag{3.38}$$

Applying Lemma 1 twice to Eq. 3.38 gives us

$$\begin{aligned}
\rho &\geq \left(1 - \sum_{i \in I_{R,X}} \frac{\mu^{1,i}_{X_0,T_0}}{2^n - r_i}\right) \cdot \left(1 - \sum_{i \in I_{X,T}} \frac{\mu^{1,i}_{R_0,X_0}}{2^n - t_i}\right) \cdot \left(1 - \sum_{i \in I_{R,X} \cup I_{X,T}} \frac{1}{2^n - x^i}\right) \\
&\geq 1 - \sum_{i \in I_{R,X}} \frac{\mu^{1,i}_{X_0,T_0}}{2^n - r_i} - \sum_{i \in I_{X,T}} \frac{\mu^{1,i}_{R_0,X_0}}{2^n - t_i} - \sum_{i \in I_{R,X} \cup I_{X,T}} \frac{1}{2^n - x^i} \\
&\geq 1 - \frac{2}{2^n} \cdot \left(\sum_{i \in I_{R,X}} \mu^{1,i}_{R_0,X_0} + \sum_{i \in I_{X,T}} \mu^{1,i}_{X_0,T_0} + nq\right). \qquad (3.39)
\end{aligned}$$

We claim that

$$\sum_{i \in I_{R,X}} \mu^{1,i}_{R_0,X_0} + \sum_{i \in I_{X,T}} \mu^{1,i}_{X_0,T_0} \leq 2nq. \qquad (3.40)$$

To see this, consider the various 4-tuples (L, R, S, T) that arise out of the oracle queries as well as the various (R, X) and (X, T) pairs that are set by the simulators. Consider the following indicator function,

$$I_{i,j,k} := \begin{cases} 1, & \text{if } R_i = R_j, T_i = T_k, X_j = X_k \\ 0, & \text{otherwise} \end{cases}$$

Here, i denotes the index of a query to $\widetilde{\pi}$, and j, k are the indices of two distinct queries to $\mathscr{S}^{\widetilde{\pi}}$. Note that

$$\sum_{i \in I_{R,X}} \mu^{1,i}_{R_0,X_0} + \sum_{i \in I_{X,T}} \mu^{1,i}_{X_0,T_0} \leq \sum_{i,j,k; j \neq k} I_{i,j,k}. \qquad (3.41)$$

The last inequality holds because any $((L, R, S, T), (R, X), (X, T))$ 3-tuple can contribute 1 to at most one of the quantities $\sum \mu^{1,i}_{R_0,X_0}$ and $\sum \mu^{1,i}_{X_0,T_0}$ depending on whether (R_0, X_0) or (X_0, T_0) appears before in the transcript. Now, the various (L_i, R_i, S_i, T_i) can come in two ways, depending on the direction of the query to $\widetilde{\pi}$. So $\sum I_{i,j,k}$ can be split into two sums according to whether the i-th oracle query was forward or backwards. Let us denote these two parts respectively by $\sum^f I_{i,j,k}$ and $\sum^b I_{i,j,k}$. Consider $\sum^f I_{i,j,k}$. Fix a pair (X, T). There can be at most n 4-tuples (L_i, R_i, S_i, T_i) which arise out of forward queries and $T_i = T$. (As $\boldsymbol{\tau}$ is a good transcript $\kappa^{\mathsf{mcoll}}_{\longrightarrow} \leq n$). For each (L_i, R_i) there can be only one (R_j, X_j) term such that $R_i = R_j$ (as X_j has to be equal to X after fixing (X, T)). So each (X, T) can contribute at most n to the sum. As there are q (X, T) pairs we have $\sum^f I_{i,j,k} \leq nq$. By fixing an (R, X) pair and following the same line of argument, it can be shown that $\sum^b I_{i,j,k} \leq nq$. Recalling Eq. 3.41 establishes the claim in Eq. 3.40.

Substituting Eq. 3.40 in Eq. 3.39 gives us

$$\rho \geq 1 - \frac{6nq}{2^n}, \qquad (3.42)$$

completing the proof of the lemma.

□

We can now use the results derived in this section to bound $\Delta_{\mathcal{A}}\left[\mathcal{G}_{1/2}, \mathcal{G}_1\right]$.

Lemma 7. *For $n \geq 6$,*

$$\Delta_{\mathcal{A}}\left[\mathcal{G}_{1/2}, \mathcal{G}_1\right] \leq \frac{n^2 q}{2^n} + \frac{6nq}{2^n} + \frac{n^2 q^2}{2^{2n+1}}.$$

Proof. The proof follows from Theorem 1, substituting $\epsilon_1 = n^2q/2^n + n^2q^2/2^{2n+1}$ (from Lemma 5) and $\epsilon_2 = 6nq/2^n$ (from Lemma 6). □

3.4.4 Main Theorem and Proof

Finally, we are ready to state and prove our main result (stated earlier in Subsec. 3.1.3).

Theorem 1 (Formal Statement)**.** *For the TLR3 construction Ψ with oracle access to E (both as described in Sec. 3.1.2, pp. 18-18), there exists a simulator $\mathscr{S}$ such that for any adversary $\mathcal{A}$ making at most q queries where $n \geq 6$, we have*

$$\Delta_{\mathcal{A}}\left[(\pi, \mathscr{S}^{\pi}), (\Psi^E, E)\right] \leq \frac{n^2 q}{2^n} + \frac{6nq}{2^n} + \frac{n^2 q^2}{2^{2n}}, \tag{3.43}$$

where π is a random permutation over $\{0,1\}^{2n}$. Further, $\mathscr{S}$ makes at most nq queries to π and has running time $\mathcal{O}(nq^2)$. Hence, Ψ^E is (t, q, ϵ)-indifferentiable from π where $t = \mathcal{O}(nq^2)$ and

$$\epsilon = \frac{n^2 q}{2^n} + \frac{6nq}{2^n} + \frac{n^2 q^2}{2^{2n}}.$$

Proof. We have

$$\Delta_{\mathcal{A}}\left[(\pi, \mathscr{S}^{\pi}), (\Psi^E, E)\right] \leq \Delta_{\mathcal{A}}\left[\mathcal{G}_0, \mathcal{G}_1\right] \leq \Delta_{\mathcal{A}}\left[\mathcal{G}_0, \mathcal{G}_{1/2}\right] + \Delta_{\mathcal{A}}\left[\mathcal{G}_{1/2}, \mathcal{G}_1\right]. \tag{3.44}$$

When $q \leq 2^n/n^2$, substituting Lemmas 4 and 7 in Eq. 3.44 establishes the bound claimed in Eq. 3.43. When $q \geq 2^n/n^2$ this bound trivially holds, since

$$\Delta_{\mathcal{A}}\left[(\pi, \mathscr{S}^{\pi}), (\Psi^E, E)\right] \leq 1 \leq \frac{n^2 q}{2^n} + \frac{6nq}{2^n} + \frac{n^2 q^2}{2^{2n}}.$$

The proof is completed by recalling the results on the efficiency of $\mathscr{S}$ discussed in Subsec. 3.2.1. □

Chapter 4

Beyond Birthday Bound Security for 5-Round Even-Mansour-Based Key-Alternating Feistel Ciphers

4.1 Introduction

In this section, we study the security of the Key-Alternating Feistel (KAF) ciphers, a class of key alternating ciphers with the Feistel structure, where each round of the cipher is instantiated with n-bit public round permutation P_i, namely the i-th round of the cipher maps

$$(X_L, X_R) \mapsto (X_R, P_i(X_R \oplus K_i) \oplus K_i \oplus X_L).$$

We have shown that our 5-round construction with independent round permutations and independent round keys achieve $2n/3$-bit security in the random permutation model, i.e., the setting where the adversary is allowed to make forward and inverse queries to the round permutations in a black box way.

4.1.1 Definition of EM-Based Key-Alternating Feistel Cipher

Given an n-bit public permutation P, and an n-bit key K, the one-round keyed Feistel permutation is the permutation on $\{0,1\}^{2n}$ is defined as follows:

$$\Psi_K^P(L\|R) = (R, L + P(R + K) + K).$$

Note that, an equivalent way of writing the above permutation $\Psi_K^P(\cdot)$ is as follows:

$$\Psi_K^P(L\|R) = (R, L + \mathsf{EM}_K^P(R)),$$

where $\mathsf{EM}_K^P(R) := P(R+K)+K$ is the one-round Even-Mansour (EM) cipher based on n-bit public round permutation P and an n-bit key K. Now, we define r-round EM-based key-alternating Feistel cipher based on r many n-bit public round permutations $\mathbf{P}^r = (P_1, P_2, \ldots, P_r) \in (\mathcal{P}_n)^r$ and a r-tuple of n-bit keys $\mathbf{K} = (K_1, K_2, \ldots, K_r) \in (\{0,1\}^n)^r$, which is denoted as $\mathsf{EM\text{-}KAF}^{\mathbf{P}^r}$. It maps an $2n$-bit plaintext $X \in \{0,1\}^{2n}$ to an $2n$-bit ciphertext as follows:

$$\mathsf{EM\text{-}KAF}_{\mathbf{K}}^{\mathbf{P}^r}(X) = \Psi_{K_r}^{P_r} \circ \Psi_{K_{r-1}}^{P_{r-1}} \circ \ldots \circ \Psi_{K_1}^{P_1}(X).$$

A pictorial description of an EM-based key-alternating cipher is shown in Fig. 4.1a.

4.2 Security Notion of EM-Based Key-Alternating Feistel Cipher

We consider distinguisher D interacting with r permutation oracles $\mathbf{P}^r = (P_1, P_2, \ldots, P_r)$, where each P_i is an n-bit random permutation, and a $2n$-bit random permutation oracle (and potentially its inverse), which is either the EM-based KAF cipher $\mathsf{EM\text{-}KAF}_{\mathbf{K}}^{\mathbf{P}^r}$ specified by a uniformly sampled $\mathbf{P}^r$ from $(\mathcal{P}_n)^r$ with a uniformly random key $\mathbf{K} = (K_1, K_2, \ldots, K_r)$ or a perfectly $2n$-bit random permutation P (independent from $\mathbf{P}^r$). We refer to $\mathsf{EM\text{-}KAF}_{\mathbf{K}}^{\mathbf{P}^r}$ / P as the construction oracle and $\mathbf{P}^r$ as the primitive oracle. We assume that the distinguisher D is adaptive, i.e., the i-th query of D is determined from the previous query-response and it is also bi-directional (i.e., it can make encryption and decryption queries to its oracles). Moreover, D is also allowed to make bi-directional queries to the primitive oracles (i.e., both forward and inverse queries) in an interleaved fashion with the construction oracle queries. We assume that D makes at most q queries to the construction oracle and at most q_i queries to the permutation oracle P_i such that $q_p = q_1 + q_2 + \ldots + q_r$. We call D to be a $(q, q_1, q_2, \ldots, q_r)$ distinguisher. We define the distinguishing advantage of D in distinguishing the outputs of the real oracle $\mathcal{O}_{\text{re}} = (\mathsf{EM\text{-}KAF}_{\mathbf{K}}^{\mathbf{P}^r}, (\mathsf{EM\text{-}KAF}_{\mathbf{K}}^{\mathbf{P}^r})^{-1}, \mathbf{P}^r)$ from the outputs of the ideal oracle $\mathcal{O}_{\text{id}} = (P, P^{-1}, \mathbf{P}^r)$ as follows:

$$\mathbf{Adv}_{\mathcal{O}_{\text{id}}}^{\mathcal{O}_{\text{re}}}(\mathsf{D}) := \Big| \Pr[\mathsf{D}^{\mathcal{O}_{\text{re}}} \Rightarrow 1] - \Pr[\mathsf{D}^{\mathcal{O}_{\text{id}}} \Rightarrow 1] \Big|, \tag{4.1}$$

where $\mathsf{D}^{\mathcal{O}} \Rightarrow 1$ denotes the event that D outputs 1 after interacting with the oracle $\mathcal{O}$. The first probability in Eqn. (4.1) is defined over the randomness of $\mathbf{K}$ and $\mathbf{P}^r$, whereas the second probability is defined over the randomness of P and $\mathbf{P}^r$. We say that $\mathsf{EM\text{-}KAF}_{\mathbf{K}}^{\mathbf{P}^r}$ is *ϵ-strong pseudorandom permutation in the random permutation model* if for all $(q, q_1, q_2, \ldots, q_r)$-distinguisher D, Eqn. (4.1) is upper bounded by ϵ. This is the security notion that we require in the paper. In the rest of the chapter, we assume that D is computationally unbounded and hence a deterministic distinguisher. We call such a distinguisher as *information theoretic distinguisher*. We also assume that D does not repeat queries and never makes pointless queries, i.e., queries whose answers can be deduced from previous query responses.

4.3 Our Contribution

All the earlier research on the security of ideal KAF ciphers is largely based on round functions and all these round functions are mostly length-preserving unkeyed functions. We know that designing pseudorandom functions is harder than designing pseudorandom permutations. Unkeyed permutations are available in plenty [12, 11, 24, 57, 47] and used in numerous Sponge-based designs [25, 24, 87, 26, 46, 40, 15]. To the best of our knowledge, there has been no prior security result on permutation-based ideal KAF ciphers. In this chapter, we study the security of an ideal KAF cipher based on unkeyed permutations. In particular, we prove that a five-round ideal KAF cipher based on five independent instances of one-round EM cipher is secure up to $O(2^{2n/3})$ queries in the random permutation model against all adversaries that are allowed to make both encryption and decryption queries to the construction. We depict existing provable security results on idealised KAF cipher in Table 4.1.

Table 4.1: Existing Provable Security Results for Ideal KAF Cipher. R denotes that the primitive is a function and P denotes that the primitive is a permutation. n denotes the domain size of the primitive. CPA denotes the adversarial model where the adversary can make only encryption queries, and CCA denotes the adversarial model where the adversary can make both encryption and decryption queries.

In the following table, we compare our result with existing security results, Rnds means rounds, and Prm means primitives, while #Rnd-Prms denotes the number of rounds a primitive has been used in the construction.

#Rnds	Key-size	Prm	#Rnd-Prms	Bound	Model	Ref
3	n	R	1	$n/2$	CPA	[93]
4	$4n$	R	2	$n/2$	CCA	[54]
4	n	R	1	$n/2$	CCA	[55]
6	$2n$	R	6	$2n/3$	CCA	[55]
12	$12n$	R	12	$2n/3$	CCA	[61]
$6t$	$6tn$	R	$6tn$	$tn/(t+1)$	CCA	[61]
5	$5n$	P	5	$2n/3$	CCA	This Paper

Remark 2. *We would like to point out here that Guo and Wang [55] show that public function-based 4-round KAF (resp. 6-round KAF) is birthday-bound (resp. beyond-birthday-bound) secure. However, the security for 5-round KAF based on public functions remains open. We believe that a 5-round KAF based on public round function can achieve beyond-birthday-bound security, and the proof should follow a similar technique as adopted in this chapter. Moreover, in the case of the public round function, we do not have to bother about the constraint that distinct inputs should map to distinct outputs, which in turn reduces both the number and the complexity of analysing the bad events. However, as there are almost no practical candidates of length preserving public round function (as they are hard to design), we chose to analyse the security of the KAF using public round permutations, which are abundance in practice (e.g., Keccak [12], SpongeNET [24], Bettle [25] etc.). We will also point out that the constructions in [**?**],[93],[54] require less key size than 5-round KAF based on public permutation. However, most of those constructions do not achieve beyond birthday security, and while the 6-round KAF based on public functions does so, it requires 6 rounds and is based on public functions rather than public permutations.*

4.4 Set up for H-Coefficient Technique

We will use the H-coefficient technique to prove the security of $\mathsf{EM\text{-}KAF}_{\mathbf{K}}^{\mathbf{P}^r}$. We consider an information-theoretic deterministic distinguisher D and two games $\mathcal{G}_1$ and $\mathcal{G}_2$ where $\mathcal{G}_1$ and $\mathcal{G}_2$ describe the distinguisher's interactions with the real world and ideal world respectively. In the real world, D interacts with the oracle $\mathcal{O}_{\text{re}} := (\mathsf{EM\text{-}KAF}_{\mathbf{K}}^{\mathbf{P}^r}, \mathbf{P}^r)$ for a uniformly chosen $\mathbf{P}^r$ from $(\mathcal{P}_n)^r$ and uniformly chosen key $\mathbf{K}$ from $(\{0,1\}^n)^r$. In the ideal world, it interacts with the oracle $\mathcal{O}_{\text{id}} := (P, \mathbf{P}^r)$, where P is a $2n$-bit to $2n$-bit uniformly sampled permutation from $\mathcal{P}_{2n}$ and $\mathbf{P}^r$ is uniformly chosen from $(\mathcal{P}_n)^r$. After this interaction is over, D outputs a decision bit $b \in \{0,1\}$. The collection of all queries and responses that are made by D to and from the oracle $\mathcal{O}$ during the interaction is summarised in a transcript (ρ, τ), where ρ summarises the overall interaction of the distinguisher D with all the primitive oracles and τ

is the transcript that summarises the interaction with the construction oracle. More formally, $\tau = \{(L_1, R_1, S_1, T_1), (L_2, R_2, S_2, T_2), \ldots, (L_q, R_q, S_q, T_q)\}$ is the set of all construction queries and responses and

$$\rho = \bigcup_{i=1}^{r} \{(U_1^i, V_1^i), (U_2^i, V_2^i), \ldots, (U_{q_i}^i, V_{q_i}^i)\}$$

is the set of all primitive queries and responses across all the primitive oracles, where we assume that D makes q construction queries and q_i for $i \in [r]$ primitive queries to the i-th primitive oracle P_i. We define for $j \in [r]$, $\mathcal{D}_j$ and ran_j be the sets of inputs and outputs of the primitive queries respectively to P_j, which we enumerate as $\mathcal{D}_j = \{U_j^1, \ldots, U_j^{q_j}\}$ and $\mathsf{ran}_j = \{V_j^1, \ldots, V_j^{q_j}\}$. Since D is bidirectional, D can make either forward construction query (L, R) and receive response (S, T) or can make inverse construction query (S, T) and receive response (L, R). Similarly, for primitive query D can either make forward query U_j^i to its primitive P_i and receive response V_j^i or can make inverse query V_j^i to P_i^{-1} and receive response U_j^i. Since we assume that D never makes pointless queries, none of the transcripts contain any duplicate elements.

We modify the experiment by releasing internal information to D after it has finished the interaction but has not yet output the decision bit. In the real world, we reveal the key $\mathbf{K} = (K_1, K_2, \ldots, K_r)$ which is used in the construction, and in the ideal world, we sample a dummy key $\mathbf{K}$ uniformly at random from $(\{0,1\}^n)^r$ and reveal it to the distinguisher. [1] In all the following, the complete transcript is $(\rho, \tau, \mathbf{K})$. Note that the modified experiment only makes the distinguisher more powerful. Hence the distinguishing advantage of D in this experiment is no way less than its distinguishing advantage in the former one.

4.5 Security Result of 5-Round EM-KAF

Here we formally state the main finding of this paper: the five-round key-alternating Feistel cipher based on Even-Mansour, which is depicted in Fig. 4.1a, and its encryption and decryption steps are listed in Fig. 4.2, is a strong pseudorandom permutation, secure against all adversaries that make $O(N^{2/3})$ construction and primitive queries in the random permutation model, where $N = 2^n$, n being the state size of the permutations and the size of the keys. We formally state this as the following theorem, the proof of which is deferred to Sec. 4.6.

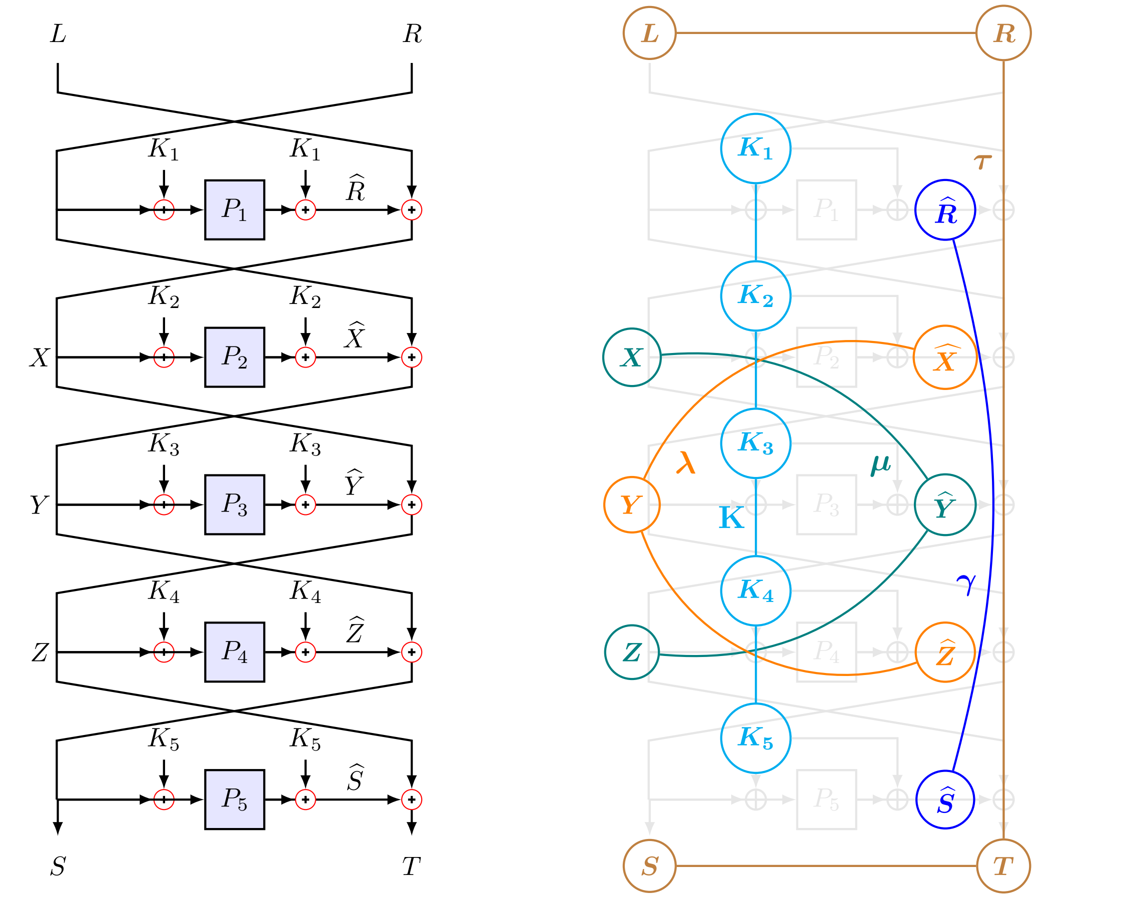
Theorem 2 (**Security Result of $\mathsf{EM\text{-}KAF}_{\mathbf{K}}^{\mathbf{P}^5}$**)**.** *Let $\mathbf{P}^5 = (P_1, P_2, P_3, P_4, P_5)$ be five independent n-bit public random permutations and $\mathbf{K} = (K_1, K_2, K_3, K_4, K_5)$ be five independent n-bit keys. Then the strong pseudorandom permutation advantage for any $(q, q_1, q_2, q_3, q_4, q_5)$-distinguisher against the construction in the random permutation model, making at most q queries to the construction and q_i primitive queries to P_i, where $q_1 + 2(\sqrt{q}+1) \leq q_2 + q_3 + q_4$, $q_5 + 2(\sqrt{q}+1) \leq q_2 + q_3 + q_4$ and $q + (q_1 + q_2 + \ldots + q_5) \leq N/2$, is given by*

$$\mathbf{Adv}^{\text{sprp-rp}}_{\mathsf{EM\text{-}KAF}_{\mathbf{K}}^{\mathbf{P}^5}}(q, q_1, \ldots, q_5) \leq \epsilon,$$

where

$$\begin{aligned}\epsilon = &\frac{6q^2}{N^2} + \frac{20q^3}{N^2} + \frac{2qq_1q_5}{N^2} + \frac{q^2}{N^2}(11q_1 + 16q_2 + 16q_3 + 16q_4 + 11q_5) + \frac{4q^4}{N^3} \\ &+ \frac{q}{N^2}(2q_1q_2 + q_1q_5 + 5q_2q_3 + 4q_2q_4 + 3q_2q_5 + 2q_1q_3 + 5q_3q_4 + 2q_3q_5 + 3q_1q_4 + 2q_4q_5) \\ &+ \frac{2q^3}{N^3}(q_1 + q_5) + \frac{q^{1/2}}{N}(q_2 + q_3 + q_4) + \frac{10q^{3/2}}{N}.\end{aligned}$$

[1] Depending on the context, oracle may also release some additional internal information.



(a) Even-Mansour Based 5-round Key-Alternating Feistel Cipher. (Diagram adapted from an example on [60].)

(b) Splitting the construction transcript into τ, K, γ, μ and γ. (The primitive transcript ρ is not shown here.)

Figure 4.1: Description of the Construction and View of The Transcript of 5-round Even-Mansour-based Key-Alternating Feistel.

The implication of the conditions $q_1 + 2(\sqrt{q}+1) \leq q_2 + q_3 + q_4$, $q_5 + 2(\sqrt{q}+1) \leq q_2 + q_3 + q_4$ is that the security holds if the total number of primitive queries to the permutation P_2, P_3 and P_4 is at least the total number of queries to permutation P_1 and the square root of the construction queries and it is also at least the total number of queries to permutation P_5 and the square root of the construction queries. With the simplifying assumption q_1, q_2, q_3, q_4 and q_5 roughly in the order of q, we have

$$\mathbf{Adv}^{\text{sprp-rp}}_{\mathsf{EM\text{-}KAF}^{\mathbf{P}^5}_{\mathbf{K}}}(q, q_1, \ldots, q_5) \quad \leq \quad \frac{6q^2}{N^2} + \frac{121q^3}{N^2} + \frac{8q^4}{N^3} + \frac{10q^{3/2}}{N}.$$

Remark 3. *From the above two conditions (i.e., $q_1+2(\sqrt{q}+1) \leq q_2+q_3+q_4$, and $q_5+2(\sqrt{q}+1) \leq q_2+q_3+q_4$), one can think about what would happen to the bound if the adversary does not make any primitive queries to the underlying permutations P_2, P_3 and P_4. We would like to mention here that we have considered an adversary that queries the underlying permutations over the* *adversary that does not. As the distinguishing advantage of the former type of adversary is always greater than the distinguishing advantage of the latter one, we bound the distinguishing advantage against the former type of adversaries that make queries to the permutations.*

Proof of Theorem 2 is the technical core of this paper. In the remainder of this section, we give an overview of our proof technique, following which the rest of the chapter is devoted to the formal proof.

```
$\mathsf{EM\text{-}KAF}_{\mathbf{K}}^{\mathbf{P}^5}(L,R)$
    $X \leftarrow P_1(R+K_1)+K_1+L$;
    $\hat{X} \leftarrow P_2(X+K_2)+K_2$;
    $Y \leftarrow \hat{X}+R$;
    $\hat{Y} \leftarrow P_3(Y+K_3)+K_3$;
    $Z \leftarrow \hat{Y}+X$;
    $\hat{Z} \leftarrow P_4(Z+K_4)+K_4$;
    $S \leftarrow \hat{Z}+Y$;
    $\hat{S} \leftarrow P_5(S+K_5)+K_5$;
    $T \leftarrow \hat{S}+Z$;
    return $(S,T)$;
```

```
$(\mathsf{EM\text{-}KAF}_{\mathbf{K}}^{\mathbf{P}^5})^{-1}(S,T)$
    $Z \leftarrow P_5(S+K_5)+K_5+T$;
    $\hat{Z} \leftarrow P_4(Z+K_4)+K_4$;
    $Y \leftarrow S+\hat{Z}$;
    $\hat{Y} \leftarrow P_3(Y+K_3)+K_3$;
    $X \leftarrow Z+\hat{Y}$;
    $\hat{X} \leftarrow P_2(X+K_2)+K_2$;
    $R \leftarrow Y+\hat{X}$;
    $\hat{R} \leftarrow P_1(R+K_1)+K_1$;
    $L \leftarrow X+\hat{R}$;
    return $(L,R)$;
```

Figure 4.2: Encryption (left) and decryption (right) algorithm of 5-round Even-Mansour Based Key-Alternating Feistel Cipher with five independent round permutations and five independent round keys.

4.5.1 Computation Order in the Real World and Transcript Notation

For each $j \in [5]$, let $\mathcal{J}_j^f$ denote the set of forward queries to P_j and $\mathcal{J}_f^b$ denote the set of backward queries to P_j, so that $\mathcal{J}_j^f \sqcup \mathcal{J}_j^b = [q_j]$. Similarly, we split the set of construction queries into the set of encryption queries $\mathcal{I}_{\mathsf{enc}}$ and the set of decryption queries $\mathcal{I}_{\mathsf{dec}}$, with $\mathcal{I}_{\mathsf{enc}} \sqcup \mathcal{I}_{\mathsf{dec}} = [q]$. For each $i \in \mathcal{I}_{\mathsf{enc}}$, the computation proceeds from the query (L^i, R^i) as shown on the left side of Fig. 4.2 to obtain (S^i, T^i), which is returned to D immediately as the response to query i, while the intermediate variables $\widehat{R}^i$, X^i, $\widehat{X}^i$, Y^i, $\widehat{Y}^i$, Z^i, $\widehat{Z}^i$, and $\widehat{S}^i$ are stored in a cache. Similarly, for each $i \in \mathcal{I}_{\mathsf{dec}}$, the computation proceeds from the query (S^i, T^i) as shown on the right side of Fig. 4.2 to obtain (L^i, R^i), which is returned to D immediately as the response to the query is stored in the cache.

For the transcript $\tau := \{(L^i, R^i, S^i, T^i) \mid i \in [q]\}$, we define the transcript *slices* $\tau^i := (L^i, R^i, S^i, T^i)$ for each $i \in [q]$, and $\tau^{\mathcal{I}} := \{\tau^i \mid i \in \mathcal{I}\}$ for each $\mathcal{I} \subseteq [q]$. At the end of the online phase, $\mathbf{K}$ is revealed to D, along with all the cached intermediate variables for each $i \in [q]$. This we call the *internal transcript*, which we split into a few parts for ease of reference. For $i \in [q]$, define $\gamma^i := (\widehat{R}^i, \widehat{S}^i)$, $\mu^i := (X^i, \widehat{Y}^i, Z^i)$, and $\lambda^i := (\widehat{X}^i, Y^i, \widehat{Z}^i)$. Analogous to τ, we define $\gamma := \{\gamma^i \mid i \in [q]\}$, $\mu := \{\mu^i \mid i \in [q]\}$, and $\lambda := \{\lambda^i \mid i \in [q]\}$ as well as the slices $\gamma^{\mathcal{I}} := \{\gamma^i \mid i \in \mathcal{I}\}$, $\mu^{\mathcal{I}} := \{\mu^i \mid i \in \mathcal{I}\}$, and $\lambda^{\mathcal{I}} := \{\lambda^i \mid i \in \mathcal{I}\}$ for each $\mathcal{I} \subseteq [q]$. The division is illustrated in Fig. 4.1b.

For each $i \in [q]$, μ^i is related to γ^i and τ^i through the equations $X^i = \widehat{R}^i + L^i = \widehat{Y}^i + Z^i$ and $Z^i = \widehat{Y}^i + X^i = \widehat{S}^i + T^i$, and λ^i is related to τ^i through the equations $Y^i = \widehat{X}^i + R^i = \widehat{Z}^i + S^i$. Thus, μ^i can be computed from τ^i and γ^i, while λ^i still retains one degree of freedom when all of τ^i, γ^i, and μ^i are fixed. Thus, in some sense, λ is the *innermost* part of the transcript, and the one that we sample at the very end in the ideal world, as described in Sec. 4.6.1.

For $\mathcal{I} \subseteq [q]$, we also define the following counting sets (along with their sizes) on the $\tau^{\mathcal{I}}$ and $\mu^{\mathcal{I}}$, which will help us in describing the ideal-world sampling mechanism in Sec. 4.6.1, as well as in analysing various sampling probabilities:

- $\mathcal{R}^{\mathcal{I}} := \{R^i \mid i \in \mathcal{I}\}$;
- $\mathcal{S}^{\mathcal{I}} := \{S^i \mid i \in \mathcal{I}\}$;
- $\mathcal{X}^{\mathcal{I}} := \{X^i \mid i \in \mathcal{I}\}$;
- $\widehat{\mathcal{Y}}^{\mathcal{I}} := \{\widehat{Y}^i \mid i \in \mathcal{I}\}$;
- $\mathcal{Z}^{\mathcal{I}} := \{Z^i \mid i \in \mathcal{I}\}$;

- $q_R^{\mathcal{I}} := |\mathcal{R}^{\mathcal{I}}|$;
- $q_S^{\mathcal{I}} := |\mathcal{S}^{\mathcal{I}}|$;
- $q_X^{\mathcal{I}} := |\mathcal{X}^{\mathcal{I}}|$;
- $q_{\widehat{Y}}^{\mathcal{I}} := |\widehat{\mathcal{Y}}^{\mathcal{I}}|$;
- $q_Z^{\mathcal{I}} := |\mathcal{Z}^{\mathcal{I}}|$.

Maintaining notational consistency with $\tau, \ldots, \lambda$, when $\mathcal{I} = [q]$ we drop the superscript and simply call the counting sets $\mathcal{R}, \ldots, \mathcal{Z}$ and their sizes $q_R, \ldots, q_Z$.

4.5.2 A Brief Overview of the Proof Strategy

We use a standard approach to bound the advantage of D with the H-Coefficient Technique. As discussed in Sec. 4.5.1, in the real world, at the end of the online phase, all the internal variables are released to D. In the ideal world, we need to *sample* these internal variables so that their distribution is close to that in the real world. Our proof hinges on this sampling mechanism, discussed at length in Sec. 4.6.1.

The basic idea behind our approach to sampling is as follows: when the online phase ends, we first sample the keys $K_1, \ldots, K_5$ randomly so that all the inputs to P_1 and P_5 are determined. We next check for collisions with $\mathcal{D}_1$ and $\mathcal{D}_5$, and mark these collision sets as $\mathcal{I}_R$ and $\mathcal{I}_S$. We also mark the queries where an R (resp. S) in the output has collided with a previous R (resp. S). The rest of the queries we bunch together as $\mathcal{I}_*$.

The next step is to sample γ. We need to do this carefully on $\mathcal{I}_*$, since if two queries have the same R (resp. S), the Y's are forced to be different, but the $\widehat{Y}$'s can collide depending on the choice of $\widehat{S}$'s (resp. $\widehat{R}$'s). For this, we arrange the queries in a tree (we can do this since we have left the collision indices out of $\mathcal{I}_*$), and sample along this tree avoiding the $\widehat{Y}$-collision described above. For the indices outside $\mathcal{I}_*$ we can choose γ randomly since a $\widehat{Y}$-collision together with the previous collisions will constitute a low probability event, which we classify as bad.

Once we have sampled γ for all indices, we can compute μ, which can be seen as one of the two internal strands. Here, we repeat what we did in the outer layer, marking all collision indices (both with primitives and among themselves) into separate sets and putting the remaining indices into $\mathcal{I}_{**}$. We avoid the same index lying in two distinct collision sets, which needs the careful bounding of a large number of bad events.

Then we come to the final step of the sampling, where we need to sample λ, maintaining consistency over P_2, P_3, and P_4. Again, the set where we need to be cautious is $\mathcal{I}_{**}$, since the consistency being accidentally violated on any of the collision sets can be classified as a bad event. Since we have kept all the collisions out of $\mathcal{I}_{**}$, we have all the μ variables distinct. Thus, the task boils down to sampling three sets of distinct variables, each of size $q_{**} = |\mathcal{I}_{**}|$, subject to $2q_{**}$ bi-variate equations. Again, we sample along the tree that was previously formed, manually avoiding collisions on any of the three variables. Outside $\mathcal{I}_{**}$, we again choose λ randomly.

The proof is then broken into two parts: bounding the probability of the bad events, and bounding the ratio of the good probabilities. The first task is long and tedious but not too challenging. For bounding the ratio of good probabilities, the challenge is to find a tight enough bound for probabilities of $\gamma^{\mathcal{I}_*}$ and $\lambda^{\mathcal{I}_{**}}$. Handling them separately does not give us a good enough bound. The key idea of the proof is the observation that the two balance each other in a way: for each previous query with the same R or same S, we have an extra constraint to take care of on γ, but we have one fewer constraint to worry about on λ, since we get the distinctness of Y for free when we ensure $\widehat{X}$ and $\widehat{Z}$ are distinct. We bank on this observation to bound the two together and successfully arrive at the desired bound.

4.6 Proof of Theorem 2

We deal with three principal components in the proof: (i) the sampling procedure in the ideal world, which enables us to define the transcript; (ii) defining and bounding the probability of bad transcripts and (iii) finally, lower bounding the ratio of the real to ideal interpolation probability for any good transcript. We begin with the sampling procedure in the ideal world in Sect. 4.6.1.

4.6.1 Sampling Procedure in the Ideal World

In the online phase, every query from D is answered with a response sampled uniformly at random from $\{0,1\}^{2n}$, as shown in Step-τa and Step-τb in Table 4.2. (We'll refer to this table throughout this section for the exact description of the sampling steps.) This leaves D with τ at the end of the online phase. Next begins the offline sampling phase of the ideal oracle, during which K_1, K_2, K_3, K_4, K_5, γ, μ and λ are sampled and released to D, such that they bear the same relations between them as their counterparts in the real world, as described in Sec. 4.5.1.

In the rest of this section, we describe step-by-step the sampling procedure in the offline phase of the ideal world. The sampling steps are intertwined with checking for several bad events. Whenever we delineate a bad event and then either resume our description of the sampling procedure or proceed to describe further bad events, we implicitly assume that we are in the scenario where the bad event was just described and all bad events described before that have not happened. Other than the usual bad events involving one or several undesirable collisions of the sampled intermediate variables either with primitive queries or between themselves, there is one specific bad event that we are keen on avoiding: for two queries i, j with $R^i = R^j$ or $S^i = S^j$, Y^i can never equal Y^j without breaking consistency with the internal relations described earlier; however, if for the same pair of queries $\widehat{R}^i + \widehat{R}^j + \widehat{S}^i + \widehat{S}^j = L^i + L^j + T^i + T^j$, $\widehat{Y}^i$ if forced to be equal to $\widehat{Y}^j$, leading to an inconsistency in P_3. We'll avoid scenarios where this can happen, and we'll indicate this by including a $\widehat{Y}$ in the name of the corresponding bad event.

Bad events on τ.

Before moving on to the online part of the sampling, we check for some bad events on τ itself. The event badτ-switch comes from the PRP-PRF switch we perform when we respond to the adversary's queries with replacement instead of without replacement, as a permutation would

Table 4.2: Sampling steps in the ideal world and the corresponding bad events that can be triggered.

Step Name	*Sampling*	*Bad Events Triggered*
Step-τa	$\forall i \in \mathcal{I}_{\mathsf{enc}}$, $(S^i, T^i) \leftarrow_\$ \{0,1\}^{2n}$	
Step-τb	$\forall i \in \mathcal{I}_{\mathsf{dec}}$, $(L^i, R^i) \leftarrow_\$ \{0,1\}^{2n}$	
		badτ-switch, badτ-$\widehat{Y}$, badτ-3path, badτ-3coll
Step-K	$\mathbf{K} \leftarrow_\$ \{0,1\}^{5n}$	
		bad$K$-outer, bad$K$-source
Step-γa	$\forall d \in [q_*]$, $\gamma_*^d \leftarrow_\$ \Gamma_*^d$	
Step-γb	$\forall S \in \mathcal{S}^{\mathcal{I}_{R*}}$, $\widehat{S} \leftarrow_\$ \{0,1\}^n$	
Step-γc	$\forall R \in \mathcal{R}^{\mathcal{I}_{S*}}$, $\widehat{R} \leftarrow_\$ \{0,1\}^n$	
		badγ-prim, badγ-coll, badγ-$\widehat{Y}$, badμ-in&out, badμ-source, badμ-inner, badμ-3coll, badμ-size
Step-λa	$\forall h \in [q_{**}]$, $\lambda_{**}^h \leftarrow_\$ \Lambda_{**}^h$	
Step-λb	$\forall X \in \mathcal{X}^{\mathcal{I}_R \sqcup \mathcal{I}_{XX}}$, $\widehat{X} \leftarrow_\$ \{0,1\}^n$	
Step-λc	$\forall Z \in \mathcal{Z}^{\mathcal{I}_S \sqcup \mathcal{I}_{ZZ}}$, $\widehat{Z} \leftarrow_\$ \{0,1\}^n$	
Step-λd	$\forall \widehat{Y} \in \widehat{\mathcal{Y}}^{\mathcal{I}_{\widehat{Y}\widehat{Y}}}$, $Y \leftarrow_\$ \{0,1\}^n$	
Step-λe	$\forall i \in \mathcal{I}_{RR} \sqcup \mathcal{I}_{SS}$, $Y^i \leftarrow_\$ \{0,1\}^n$	
		badλ-prim, badλ-coll

do. The event badτ-$\widehat{Y}$ is the forced collision on $\widehat{Y}$ we mentioned earlier. badτ-3path involves a simultaneous 3-collision on R and S, which must involve a *path* of length 3. (For instance, one way to achieve this is as follows: an encryption query (L_1, R) giving (S, T_1); then a decryption query (S, T_2) yielding (L_2, R), making a path of length 2; and finally, a second encryption query with (L_3, R) giving (S, T_3), extending the path to length 3.) Finally, the event badτ-3coll involves a 3-collision on R or S where the last two come from oracle outputs. The precise definitions of these bad events are given in Fig. 4.3.

Sampling K and bad events thereof.

Once none of the bad events on τ has happened, we move on to the offline phase of the sampling. Let $\mathcal{I}_{RR} := \{i \in \mathcal{I}_{\mathsf{dec}} \mid R^i = R^j \text{ for some } j \in [i-1]\}$ and $\mathcal{I}_{SS} := \{i \in \mathcal{I}_{\mathsf{enc}} \mid S^i = S^j \text{ for some } j \in [i-1]\}$ be the index sets where an R or S obtained from an oracle response collides with a previously seen one (either as part of a query or as part of a response).

The first step in the offline phase is to sample the keys K_1, K_2, K_3, K_4, and K_5 independently and uniformly at random from $\{0,1\}^n$. This determines all the inputs to P_1 and P_5. We define the index sets $\mathcal{I}_R := \{i \in [q] \mid R^i + K_1 \in \mathcal{D}_1\}$ and $\mathcal{I}_S := \{i \in [q] \mid S^i + K_5 \in \mathcal{D}_5\}$, where the outputs of P_1 and P_5 are already determined from ρ.

Figure 4.3: badτ

badτ-switch: $\exists i, j \in [q], i < j, [j \in \mathcal{I}_{\mathsf{enc}}, (S^i, T^i) = (S^j, T^j)] \vee [j \in \mathcal{I}_{\mathsf{dec}}, (L^i, R^i) = (L^j, R^j)]$.

badτ-$\widehat{Y}$: $\exists i, j \in [q], i < j, [R^i = R^j] \wedge [S^i = S^j] \wedge [L^i + T^i = L^j + T^j]$.

badτ-3path: $\exists\, i, j, l \in [q], i < j < l, [R^i = R^j = R^l] \wedge [S^i = S^j = S^l]$.

badτ-3coll: $\exists\, i, j, l \in [q], i < j < l, [j, l \in \mathcal{I}_{\mathsf{dec}}, R^i = R^j = R^l] \vee [j, l \in \mathcal{I}_{\mathsf{enc}}, S^i = S^j = S^l]$.

Figure 4.4: bad$K$

bad$K$-outer: $\mathcal{I}_R$, $\mathcal{I}_{RR}$, $\mathcal{I}_S$, and $\mathcal{I}_{SS}$ are not pairwise disjoint.

bad$K$-source: $\exists i, j \in [q], i < j, [i \in \mathcal{I}_S \sqcup \mathcal{I}_{SS}, j \in \mathcal{I}_{RR}, R^i = R^j] \vee [i \in \mathcal{I}_R \sqcup \mathcal{I}_{RR}, j \in \mathcal{I}_{SS}, S^i = S^j]$.

Sampling the keys can trigger two bad events: bad$K$-outer is the event when an encryption query index lies in two of the sets $\mathcal{I}_R$, $\mathcal{I}_S$, and $\mathcal{I}_{SS}$ at the same time, or a decryption query index lie in two of the sets $\mathcal{I}_R$, $\mathcal{I}_S$, and $\mathcal{I}_{RR}$ at the same time; and bad$K$-source, where the *source* of a collision index in $\mathcal{I}_{RR}$ (resp. $\mathcal{I}_{SS}$) (the earlier R (resp. S) value where it collided) lies in one of $\mathcal{I}_R$, $\mathcal{I}_S$, and $\mathcal{I}_{SS}$ (resp. $\mathcal{I}_{RR}$). The definitions can be found in Fig. 4.4.

Defining and computing $G[\tau_*]$.

When sampling γ, we begin with $\mathcal{I}_*$. Since queries in $\mathcal{I}_*$ do not come from another collision event, we need to avoid bad collision events manually while sampling $\gamma^{\mathcal{I}_*}$.

Define $\tau_* := \tau^{\mathcal{I}_*}$, $\mathcal{R}_* := \mathcal{R}^{\mathcal{I}_*}$, $\mathcal{S}_* := \mathcal{S}^{\mathcal{I}_*}$. Consider the directed bipartite graph $G[\tau_*]$ with vertices in $\mathcal{R}_*$ and $\mathcal{S}_*$, where we put an edge between $R \in \mathcal{R}_*$ and $S \in \mathcal{S}_*$ if there is a query $i \in \mathcal{I}_*$ with $R^i = R$ and $S^i = S$; the direction of the edge is from R to S if $i \in \mathcal{I}_{\mathsf{enc}}* := \mathcal{I}_{\mathsf{enc}} \cap \mathcal{I}_*$ and S to R if $i \in \mathcal{I}_{\mathsf{dec}}* := \mathcal{I}_{\mathsf{dec}} \cap \mathcal{I}_*$.

Since we are in $\mathcal{I}_*$, we know that there are no cycles in $G[\tau_*]$, making it a forest. Let M be the number of trees in $G[\tau_*]$. Define $q_* := |\mathcal{I}_*|$, $q_{R*} := |\mathcal{R}_*|$, $q_{S*} := |\mathcal{S}_*|$. Since $G[\tau_*]$ has $q_{S*} + q_{R*}$ vertices and q_* edges, we have

$$q_{R*} + q_{S*} = q_* + M. \tag{4.2}$$

We observe further that a new tree is added to this forest exactly on each query in the set $\{i \in \mathcal{I}_{\mathsf{enc}}* \mid R^i \notin \mathcal{R}^{[i-1]}\} \sqcup \{i \in \mathcal{I}_{\mathsf{dec}}* \mid S^i \notin \mathcal{S}^{[i-1]}\}$, i.e., on each encryption query in $\mathcal{I}_*$ with

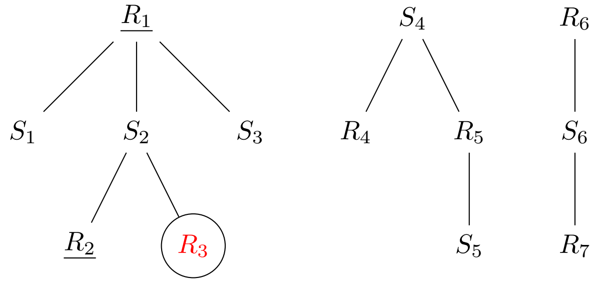


Figure 4.5: The forest structure on $\mathcal{I}_*$. For instance, the node R_3 (here circled) represents a decryption query (S_2, T) for some T, that outputs (L, R_3) for some L. This is the first query where R_3 appears, and to count the number of earlier queries in which S_2 appears, we only need to look at this node's grandparent and elder siblings (R_1 and R_2 respectively, here underlined).

a fresh R and each decryption query in $\mathcal{I}_*$ with a fresh S; we call the resulting trees R-rooted (with root R^i) and S-rooted (with root S^i) respectively.

We label $\mathcal{R}_*$ and $\mathcal{S}_*$ as follows: first, the trees are arranged in query order of the roots; next, within each tree, we begin with the root and do a breadth-first traversal, discovering R-generations and S-generations alternately. Finally, we order $\mathcal{R}_*$ and $\mathcal{S}_*$ separately, first by trees, then within the same tree by generations, then within the same generation by parents' order, and finally among siblings by order of appearance. This gives us a total order on both $\mathcal{R}_*$ and $\mathcal{S}_*$, and allow us to label them $R_1, \ldots, R_{q_{R*}}$ and $S_1, \ldots, S_{q_{S*}}$ respectively. We also extend the notation $\widehat{R}_\ell := \widehat{R}^i$ for i such that $R_\ell = R^i$, and $\widehat{S}_m := \widehat{S}^i$ for i such that $S_m = S^i$.

We will also find it convenient to refer to the queries by the end-labels of the edge they correspond to: a query $i \in \mathcal{I}_{\mathsf{enc}}*$ with $R^i = R_\ell$ and $S^i = S_m$ will be referred to as (ℓ, m), while a query $i \in \mathcal{I}_{\mathsf{dec}}*$ with $S^i = S_m$ and $R^i = R_\ell$ will be referred to as (m, ℓ). We order the queries as follows: two encryption queries (ℓ, m) and (ℓ', m') have the same order as m and m', while two decryption queries (m, ℓ) and (m', ℓ') have the same order as ℓ and ℓ'; finally, to compare an encryption query (ℓ, m) and a decryption query (m', ℓ') we note that they must be either in different trees, or in different generations of the same tree, and order them as we ordered the vertices in the corresponding cases. Fig. 4.5 illustrates the forest structure.

For each $i \in \mathcal{I}_*$, let d_i denote the rank of i in the new ordering. Then $i \mapsto d_i$ is a bijection from $\mathcal{I}_*$ to $[q_*]$. We'll use $d = d_i$ interchangeably with the end-labels (ℓ, m) or (m, ℓ) to refer to a query in $\mathcal{I}_*$. We write ℓ^d and m^d to denote the end-labels of d, irrespective of the direction of the query. (Note that we'll often write rank to mean the rank of some node in this ordering; it is not to be confused with the rank of a matrix.)

Sampling γ.

Before sampling γ, we set the values already determined from primitive collisions: for each $i \in \mathcal{I}_R$ we set $\widehat{R}^i \leftarrow V_1^j + K_1$ where j is such that $U_1^j = R^i + K_1$, and for each $i \in \mathcal{I}_S$ we set $\widehat{S}^i \leftarrow V_5^j + K_5$ where j is such that $U_5^j = S^i + K_5$. Using the graph $G[\tau_*]$, we describe a sampling mechanism for $\gamma^{\mathcal{I}_*}$. For $\mathcal{I} \subseteq \mathcal{I}_*$ we call a $\gamma^{\mathcal{I}}$ *valid* if it satisfies the following conditions:

- $\widehat{R}^i + K_1 \notin \mathsf{ran}_1$ for each $i \in \mathcal{I} \setminus \mathcal{I}_R$;
- $\widehat{S}^i + K_5 \notin \mathsf{ran}_5$ for each $i \in \mathcal{I} \setminus \mathcal{I}_S$;

and for each distinct $i, j \in \mathcal{I}$:

- $R^i = R^j \iff \widehat{R}^i = \widehat{R}^j$;
- $S^i = S^j \iff \widehat{S}^i = \widehat{S}^j$;
- $R^i = R^j \implies \widehat{S}^i + \widehat{S}^j \neq L^i + T^i + L^j + T^j$;
- $S^i = S^j \implies \widehat{R}^i + \widehat{R}^j \neq L^i + T^i + L^j + T^j$.

Let $d_{\mathcal{I}} := \{d_i \mid i \in \mathcal{I}\}$. Let $\gamma_*^{d_i} := \gamma^i$ for each $i \in \mathcal{I}_*$, and $\gamma_*^{d_{\mathcal{I}}} := \gamma^{\mathcal{I}}$ for any $\mathcal{I} \subseteq \mathcal{I}_*$. Let Γ_{good} be the set $\{\gamma^{\mathcal{I}} \mid \mathcal{I} \subseteq \mathcal{I}_*, \gamma^{\mathcal{I}} \text{ is valid}\}$. Given a $\gamma_*^{[d-1]} \in \Gamma_{\mathsf{good}}$, let $\Gamma_*^d := \Gamma_*^d[\gamma_*^{[d-1]}]$ be the set of values γ_*^d can take such that $\gamma_*^{[d]}$ remains in Γ_{good}. We note that unless the edge corresponding to query d begins in a root node, one half of γ_*^d will already be fixed from $\gamma_*^{[d-1]}$. For instance, for a query (ℓ^d, m^d) with a non-root source R_{ℓ^d}, there is a previous query (m^c, ℓ^c) with $c < d$ such that $R_{\ell^c} = R_{\ell^d}$, so $\widehat{R}_{\ell^d}$ is determined from γ_*^c. For this case, each value in Γ_*^d will look like $(\widehat{R}_{\ell^c}, \widehat{S})$ for some candidate value $\widehat{S}$ for $\widehat{S}_{m^d}$.

Then we sample $\gamma^{\mathcal{I}_*} = \gamma_*^{[q_*]}$ as follows: for each $d \in [q*]$, having sampled $\gamma_*^{[d-1]}$, we sample γ_*^d uniformly at random from Γ_*^d. This is shown as Step-γa in Table 4.2. Then we proceed to compute the index sets $\mathcal{I}_{R*} := \{i \in \mathcal{I}_R \sqcup \mathcal{I}_{RR} \mid S^i \notin \mathcal{S}_*\}$ and $\mathcal{I}_{S*} := \{i \in \mathcal{I}_S \sqcup \mathcal{I}_{SS} \mid R^i \notin \mathcal{R}_*\}$. Finally, for each $S \in \mathcal{S}^{\mathcal{I}_{R*}}$ (resp. $R \in \mathcal{R}^{\mathcal{I}_{S*}}$), we sample $\widehat{S}$ (resp. $\widehat{R}$) uniformly at random from $\{0,1\}^n$, as shown in Step-γb (resp. Step-γc) in Table 4.2. This completes our sampling of γ.

Figure 4.6: badγ

badγ-prim: $\exists i \in [q]$, $[i \in \mathcal{I}_R^c, \widehat{R}^i + K_1 \in \mathsf{ran}_1] \vee [i \in \mathcal{I}_S^c, \widehat{S}^i + K_5 \in \mathsf{ran}_5]$.

badγ-coll: $\exists i,j \in [q], i < j$, $[[R^i \neq R^j] \wedge [\widehat{R}^i = \widehat{R}^j]] \vee [[S^i \neq S^j] \wedge [\widehat{S}^i = \widehat{S}^j]]$.

badγ-$\widehat{Y}$: $\exists i,j \in [q], i < j$, $[[R^i = R^j] \wedge [\widehat{S}^i + \widehat{S}^j = L^i + T^i + L^j + T^j]] \vee [[S^i = S^j] \wedge [\widehat{R}^i + \widehat{R}^j = L^i + T^i + L^j + T^j]]$.

Bad events on γ.

The bad events on γ come from evaluating the conditions for $\gamma^{\mathcal{I}_*}$ being valid on the entire γ. badγ-prim arises from a primitive collision outside on the range of P_1 (resp. P_5) outside $\mathcal{I}_R$ (resp. $\mathcal{I}_S$). badγ-coll is the event of a collision of $\widehat{R}$ (resp. $\widehat{S}$) on two distinct values of R (resp. S). Finally, badγ-$\widehat{Y}$ is the event of a collision on $\widehat{R} + \widehat{S} + L + T$ on two queries with the same R or same S (both of which forces Y to be distinct on these two queries). The definitions can be found in Fig. 4.6.

Figure 4.7: badμ

badμ-in&out: $\mathcal{I}_{\text{outer}} \cap \mathcal{I}_{\text{inner}} \neq \emptyset$.

badμ-source: $\exists i, j \in [q]$, $[i \in \mathcal{I}_R, j \in \mathcal{I}_{XX}, X^i = X^j] \vee [i \in \mathcal{I}_S, j \in \mathcal{I}_{ZZ}, Z^i = Z^j]$.

badμ-inner: $\mathcal{I}_X$, $\mathcal{I}_{XX}$, $\mathcal{I}_{\widehat{Y}}$, $\mathcal{I}_{\widehat{Y}\widehat{Y}}$, $\mathcal{I}_Z$ and $\mathcal{I}_{ZZ}$ are not pairwise disjoint.

badμ-3coll: $\exists i, j, l \in \mathcal{I}_{\text{inner}}, i < j < l$, $[X^i = X^j = X^l] \vee [\widehat{Y}^i = \widehat{Y}^j = \widehat{Y}^l] \vee [Z^i = Z^j = Z^l]$.

badμ-size: $|\mathcal{I}_{\text{inner}}| > \sqrt{q}$.

Bad events on μ.

Next, we compute μ from τ and γ using the equations in Sec. 4.5.1. Define the collision sets $\mathcal{I}_X := \{i \in [q] \mid X^i + K_2 \in \mathcal{D}_2\}$, $\mathcal{I}_{\widehat{Y}} := \{i \in [q] \mid \widehat{Y}^i + K_3 \in \mathsf{ran}_3\}$, $\mathcal{I}_Z := \{i \in [q] \mid Z^i + K_4 \in \mathcal{D}_4\}$, $\mathcal{I}_{XX} := \{i \in \mathcal{I}_R^c \mid X^i = X^j \text{ for some } j \in [q]\}$, $\mathcal{I}_{\widehat{Y}\widehat{Y}} := \{i \in [q] \mid \widehat{Y}^i = \widehat{Y}^j \text{ for some } j \in [q]\}$, $\mathcal{I}_{ZZ} := \{i \in \mathcal{I}_S^c \mid Z^i = Z^j \text{ for some } j \in [q]\}$. Further define $\mathcal{I}_{\text{outer}} := \mathcal{I}_R \cup \mathcal{I}_{RR} \cup \mathcal{I}_S \cup \mathcal{I}_{SS}$ and $\mathcal{I}_{\text{inner}} := \mathcal{I}_X \cup \mathcal{I}_{XX} \cup \mathcal{I}_{\widehat{Y}} \cup \mathcal{I}_{\widehat{Y}\widehat{Y}} \cup \mathcal{I}_Z \cup \mathcal{I}_{ZZ}$, and $\mathcal{I}_{**} := \mathcal{I}_* \setminus \mathcal{I}_{\text{inner}}$. The event bad$\mu$-in&out occurs when one of the outer collision indices in $\mathcal{I}_{\text{outer}}$ is also in $\mathcal{I}_{\text{inner}}$. The event bad$\mu$-inner occurs when an index lies at once in two inner collision sets $\mathcal{I}_X$, $\mathcal{I}_{XX}$, $\mathcal{I}_{\widehat{Y}}$, $\mathcal{I}_{\widehat{Y}\widehat{Y}}$, $\mathcal{I}_Z$ and $\mathcal{I}_{ZZ}$. badμ-source checks for a collision index in $\mathcal{I}_{XX}$ (resp. $\mathcal{I}_{ZZ}$) with its source index in $\mathcal{I}_R$ (resp. $\mathcal{I}_S$). (Note that unlike in bad$K$-source, the query order of these two indices is not important here.) badμ-3coll captures 3-collisions on any of the variables X, $\widehat{Y}$ or Z. Finally, badμ-size is the event that the set of inner collisions grows too big. The definitions can be found in Fig. 4.7.

Sampling λ.

Before sampling λ, we set the values already determined from primitive collisions: for each $i \in \mathcal{I}_X$ we set $\widehat{X}^i \leftarrow V_2^j + K_2$ where j is such that $U_2^j = X^i + K_2$; for each $i \in \mathcal{I}_{\widehat{Y}}$ we set $Y^i \leftarrow V_3^j + K_3$ where j is such that $U_3^j = \widehat{Y}^i + K_3$, and for each $i \in \mathcal{I}_Z$ we set $\widehat{Z}^i \leftarrow V_4^j + K_4$ where j is such that $U_4^j = Z^i + K_4$. To describe a sampling mechanism for $\lambda^{\mathcal{I}_{**}}$, we return to the graph $G[\tau_*]$. For $\mathcal{I} \subseteq \mathcal{I}_{**}$ we call a $\lambda^{\mathcal{I}}$ *valid* if it satisfies the following conditions:

- $\widehat{X}^i + K_2 \notin \mathsf{ran}_2$ for each $i \in \mathcal{I} \setminus \mathcal{I}_X$;
- $Y^i + K_3 \notin \mathcal{D}_3$ for each $i \in \mathcal{I} \setminus \mathcal{I}_{\widehat{Y}}$;
- $\widehat{Z}^i + K_4 \notin \mathsf{ran}_4$ for each $i \in \mathcal{I} \setminus \mathcal{I}_Z$.
- $\widehat{X}^i + Y^i = R^i$ for each $i \in \mathcal{I}$;

- $Y^i + \widehat{Z}^i = S^i$ for each $i \in \mathcal{I}$;

and for each distinct $i, j \in \mathcal{I}$:

- $X^i = X^j \iff \widehat{X}^i = \widehat{X}^j$;
- $\widehat{Y}^i = \widehat{Y}^j \iff Y^i = Y^j$;
- $Z^i = Z^j \iff \widehat{Z}^i = \widehat{Z}^j$.

Define $q_{**} := |\mathcal{I}_{**}|$. Suppose we take the relabeled queries $1, \ldots, q_*$, drop the queries pertaining to $\mathcal{I}_* \setminus \mathcal{I}_{**}$, and renumber the remaining indices $1, \ldots, q_{**}$. We call h_i the index of query i under this new renumbering. Thus, h_i is obtained by subtracting from d_i the number of queries in $[d_i - 1]$ that come from outside $\mathcal{I}_{**}$. Let $h_{\mathcal{I}} := \{h_i \mid i \in \mathcal{I}\}$. Let $\lambda_{**}^{h_i} := \lambda^i$ for any $i \in \mathcal{I}_{**}$, and $\lambda_{**}^{h_{\mathcal{I}}} := \lambda^{\mathcal{I}}$ for any $\mathcal{I} \subseteq \mathcal{I}_{**}$. Let Λ_{good} be the set $\{\lambda^{\mathcal{I}} \mid \mathcal{I} \subseteq \mathcal{I}_{**}, \lambda^{\mathcal{I}} \text{ is valid}\}$. Given a $\lambda_{**}^{[h-1]} \in \Lambda_{\mathsf{good}}$, let $\Lambda_{**}^h := \Lambda_{**}^h[\lambda_{**}^{[h-1]}]$ be the set of values λ_{**}^h can take such that $\lambda_{**}^{[h]}$ remains in Λ_{good}.

Then we sample $\lambda^{\mathcal{I}_{**}} = \lambda_{**}^{[q_{**}]}$ as follows: for each $h \in [q_{**}]$, having sampled $\lambda_{**}^{[h-1]}$, we sample λ_{**}^h uniformly at random from Λ_{**}^h. This is shown as Step-λa in Table 4.2. Sampling the rest of λ is straightforward: for each distinct X on $\mathcal{I}_R \sqcup \mathcal{I}_{XX}$, $\widehat{X}$ is sampled uniformly at random from $\{0,1\}^n$ (Step-λb); and we similarly sample $\widehat{Z}$ for each distinct Z on $\mathcal{I}_S \sqcup \mathcal{I}_{ZZ}$ (Step-λc) and Y for each distinct $\widehat{Y}$ on $\mathcal{I}_{\widehat{Y}\widehat{Y}}$ (Step-λd). Finally, for each query in $\mathcal{I}_{RR} \sqcup \mathcal{I}_{SS}$, we sample Y^i uniformly at random. Since fixing one of the variables in λ^i determines the other two, this completes the sampling of λ, and brings us to the end of our sampling procedure.

Figure 4.8: badλ

badλ-prim: $\exists i \in [q]$, $[i \in \mathcal{I}_X^c, \widehat{X}^i + K_2 \in \mathsf{ran}_2] \vee [i \in \mathcal{I}_{\widehat{Y}}^c, Y^i + K_3 \in \mathcal{D}_3] \vee [i \in \mathcal{I}_Z^c, \widehat{Z}^i + K_4 \in \mathsf{ran}_4]$.

badλ-coll: $\exists i, j \in [q], i < j$, $[[X^i \neq X^j] \wedge [\widehat{X}^i = \widehat{X}^j]] \vee [[\widehat{Y}^i \neq \widehat{Y}^j] \wedge [Y^i = Y^j]] \vee [[Z^i \neq Z^j] \wedge [\widehat{Z}^i = \widehat{Z}^j]]$.

Bad events on λ.

The bad events on λ come from evaluating the conditions for $\lambda^{\mathcal{I}_*}$ being valid on the entire λ. badλ-prim arises from a primitive collision outside on the range of P_2 (resp. domain of P_3; range of P_4) outside $\mathcal{I}_X$ (resp. $\mathcal{I}_{\widehat{Y}}$; $\mathcal{I}_Z$). badλ-coll is the event of a collision of $\widehat{X}$ (resp. Y; $\widehat{Z}$) on two distinct values of X (resp. $\widehat{Y}$; Z). The definitions can be found in Fig. 4.8.

Definition of Bad Transcripts, Bad Lemma and Good Lemma.

In this sampling procedure, if none of the above bad events happens, we release all the internal variables, i.e., γ, μ, λ and the round keys $(K_1, K_2, K_3, K_4, K_5)$ along with the input-output query responses (L, R, S, T) to the adversary. After the interaction is over with the construction oracle and the primitive oracles, we summarise the interaction in a *transcript* that records all the input outputs of the interaction along with the corresponding internal variables, i.e,

$\eta = (\rho, \tau, \mathbf{K}, \gamma, \mu, \lambda)$, where $\tau = \{(L^i, R^i, S^i, T^i) : i \in [q]\}$ and $\rho = \{(U_1^i, V_1^i), (U_2^i, V_2^i), \ldots, (U_{q_i}^i, V_{q_i}^i) : i \in [5]\}$, where U_j^i (resp. V_j^i) is the j-th primitive input (resp. primitive output) to the i-th permutation P_i.

Definition 6 (**Bad Transcript**)**.** *A transcript $\eta = (\rho, \tau, \mathbf{K}, \gamma, \mu, \lambda)$ is said to be* bad *if any of the above bad events i.e., badτ, badK, badγ, badμ, badλ happen.*

Lemma 8 (**Bad Lemma**)**.** *Let $\eta = (\rho, \tau, \mathbf{K}, \gamma, \mu, \lambda)$ be any attainable transcript. Then we bound the proability of obtaining* bad *in the game $\mathcal{G}_2$ by the following lemma:*

$$\begin{aligned}\Pr[\eta \in \mathsf{bad}] \quad \leq \quad & \frac{6q^2}{N^2} + \frac{14q^3}{N^2} + \frac{4q^4}{N^3} + \frac{2q^3}{N^3}(q_1 + q_5) + \frac{q^{1/2}}{N}(q_2 + q_3 + q_4) + \frac{2q^{3/2}}{N} \\ & + \frac{2qq_1q_5}{N^2} + \frac{q^2}{N^2}(11q_1 + 12q_2 + 12q_3 + 12q_4 + 11q_5) \\ & + \frac{q}{N^2}(2q_1q_2 + q_1q_5 + 3q_2q_3 + 2q_2q_4 + 3q_2q_5 + 2q_1q_3 + 3q_3q_4 \\ & + 2q_3q_5 + 3q_1q_4 + 2q_4q_5).\end{aligned}$$

By assuming q_1, q_2, q_3, q_4 and q_5 roughly in the order of q, then we have

$$\Pr[\eta \in \mathsf{bad}] \quad \leq \quad \frac{6q^2}{N^2} + \frac{97q^3}{N^2} + \frac{8q^4}{N^3} + \frac{5q^{3/2}}{N}.$$

This lemma is proved by an exhaustive case-by-case analysis of all the listed bad events and all possible sub-events that give rise to them. The trickiest part of the proof is to bound the probability of badγ, which is given below. The rest of the proof is more straight-forward and deferred to section 4.8

4.6.2 Bounding badγ-prim

Proposition 1. *Having defined the bad event badγ-prim in Fig. 4.6, we have*

$$\Pr[\mathsf{bad}\gamma\text{-}\mathsf{prim}] \leq \frac{qq_5(q_1 + q_2)}{N^2} + \frac{(q_1 + q_5)\binom{q}{2}}{N^2}.$$

Now, to bound badγ-prim, we further split it into the following two cases:

- badγ-prim-1. $\exists i \in \mathcal{I}_{R*}$ and $j \in [q_5]$ such that $\widehat{S}^i + K_1 = V_5^j$.
- badγ-prim-2. $\exists i \in \mathcal{I}_{S*}$ and $j \in [q_1]$ such that $\widehat{R}^i + K_1 = V_1^j$.

Bounding badγ-prim-1.

We split the event into the following sub-cases and bound the probabilities of each of them.

- badγ-prim-1a. $\exists i \in \mathcal{I}_{R*} \cap \mathcal{I}_R$ and $j \in [q_5]$ such that $\widehat{S}^i + k_5 = V_5^j$.

  In other words, $\exists i \in q$, $j \in [q_5]$ and $l \in [q_2]$ such that $R^i + K_1 = U_1^l$ and $\widehat{S}^i + K_5 = V_5^j$. Let's first fix the values for the indices i, j and l. The probability of each of the events comes out to be $(1/N)$ due to the n-bit randomness over the keys K_1 and K_5, respectively. As we can choose the indices i, j, and l in q, q_5, and q_2 ways, we use the union bound over all those possible choices to obtain

$$\Pr[\mathsf{bad}\gamma\text{-}\mathsf{prim}\text{-}1a] \leq \frac{qq_2q_5}{N^2}\,. \tag{4.3}$$

- badγ-prim-1b. $\exists i \in \mathcal{I}_{R*} \cap \mathcal{I}_{RR}$ and $j \in [q_5]$ such that $\widehat{S}^i + K_5 = V_5^j$.

  In other words, $\exists i \in \mathcal{I}_{\mathsf{dec}}$, $j \in [q_5]$ and $l \in [i-1]$ such that $R^i = R^l$ and $\widehat{S}^i + K_5 = V_5^j$. Let's first fix the values for the indices i, j and l. The probability of the event $R^i = R^l$ comes out to be $(1/N)$ due to the n-bit randomness over R^i as $i > l$ and $i \in \mathcal{I}_{\mathsf{dec}}$. The probability of the event $\widehat{S}^i + K_5 = V_5^j$ comes out to be $(1/N)$ due to the n-bit randomness over the key K_5. As we can choose the pair of indices (i, l) in $\binom{q}{2}$ ways and the index j in q_5 ways, we use the union bound over all those possible choices to obtain

$$\Pr[\text{bad}\gamma\text{-prim-1}b] \leq \frac{q_5\binom{q}{2}}{N^2}\,. \tag{4.4}$$

Adding the probabilities of the above two cases, we obtain

$$\Pr[\text{bad}\gamma\text{-prim-1}] \leq \frac{qq_2q_5}{N^2} + \frac{q_5\binom{q}{2}}{N^2}\,. \tag{4.5}$$

Bounding badγ-prim-2.

As before, we split the event into the following sub-cases and bound the probabilities of each of them.

- badγ-prim-2a. $\exists i \in \mathcal{I}_{S*} \cap \mathcal{I}_{S}$ and $j \in [q_1]$ such that $\widehat{R}^i + K_1 = V_1^j$.

  In other words, $\exists i \in q$, $j \in q_1$ and $l \in q_2$ such that $S^i + K_5 = V_5^l$ and $\widehat{R}^i + K_1 = V_1^j$. Let's first fix the values for the indices i, j and l. The probability of each of the events comes out to be $(1/N)$ due to the n-bit randomness over the keys K_1 and K_5 respectively. As we can choose the indices i, j and l in q, q_5 and q_1 ways, we use the union bound over all those possible choices to obtain

$$\Pr[\text{bad}\gamma\text{-prim-2}a] \leq \frac{qq_1q_5}{N^2}\,. \tag{4.6}$$

- badγ-prim-2b. $\exists i \in \mathcal{I}_{S*} \cap \mathcal{I}_{SS}$ and $j \in [q_1]$ such that $\widehat{R}^i + K_1 = V_1^j$.

  In other words, $\exists i \in \mathcal{I}_{\mathsf{enc}}$, $j \in [q_1]$ and $l \in [i-1]$ such that $S^i = S^l$ and $\widehat{R}^i + K_1 = V_1^j$. Let's first fix the values for the indices i, j and l. The probability of the event $S^i = S^l$ comes out to be $(1/N)$ due to the n-bit randomness over S^i as $i > l$ and $i \in \mathcal{I}_{\mathsf{enc}}$. The probability of the event $\widehat{R}^i + K_1 = V_1^j$ comes out to be $(1/N)$ due to the n-bit randomness over the key K_1. As we can choose the pair of indices (i, l) in $\binom{q}{2}$ ways and the index j in q_1 ways, we use the union bound over all those possible choices to obtain

$$\Pr[\text{bad}\gamma\text{-prim-2}b] \leq \frac{q_1\binom{q}{2}}{N^2}\,. \tag{4.7}$$

Adding the probabilities of the above two cases, we obtain

$$\Pr[\text{bad}\gamma\text{-prim-2}] \leq \frac{qq_1q_5}{N^2} + \frac{q_1\binom{q}{2}}{N^2}\,. \tag{4.8}$$

By combining Eqn. (4.5) and Eqn. (4.8), we have

$$\Pr[\text{bad}\gamma\text{-prim}] \leq \frac{qq_5(q_1+q_2)}{N^2} + \frac{(q_1+q_5)\binom{q}{2}}{N^2}\,. \tag{4.9}$$

4.6.3 Bounding badγ-coll

Proposition 2. *Having defined the bad event badγ-coll in Fig. 4.6, we have*

$$\Pr[\mathsf{bad}\gamma\text{-}\mathsf{coll}] \le \frac{q^2(q_1+q_5)}{N^2} + \frac{4q^4}{N^3} + \frac{2q^3(q_1+q_5)}{N^3}.$$

As before, to bound badγ-coll, we further split it into the following two cases:

- badγ-coll-1. $\exists i,j \in \mathcal{I}_{R*}$ and $i \ne j$ such that $S^i \ne S^j$ and $\widehat{S}^i = \widehat{S}^j$.
- badγ-coll-2. $\exists i,j \in \mathcal{I}_{S*}$ and $i \ne j$ such that $R^i \ne R^j$ and $\widehat{R}^i = \widehat{R}^j$.

Bounding badγ-coll-1.

As before, we split the event into the following sub-cases and bound the probabilities of each of them.

- badγ-coll-1a. $\exists i,j \in \mathcal{I}_{R*} \cap \mathcal{I}_R$ and $i \ne j$ such that $S^i \ne S^j$ and $\widehat{S}^i = \widehat{S}^j$.
  In other words, $\exists i,j \in \mathcal{I}_R$, such that $i \ne j$, and $k,l \in [q_1]$ such that
  $$R^i + K_1 = U_1^k, R^j + K_1 = U_1^l, \widehat{S}^i = \widehat{S}^j.$$
  We can write the above event in an equivalent way as
  $$R^i + K_1 = U_1^k, R^i + R^j = U_1^k + U_1^l, \widehat{S}^i = \widehat{S}^j.$$
  Let's first fix the values for the indices i, j, k and l and without loss of generality, we assume that $i > j$. The probability of the event $R^i + K_1 = U_1^k$ comes out to be $(1/N)$ due to the n-bit randomness over the key K_1. Moreover, the probability of the event $\widehat{S}^i = \widehat{S}^j$ comes out to be at most $2/N$ due to the randomness of $\widehat{S}^i$. However, the number of choices of indices (i,j,k,l) such that $R^i + R^j = U_1^k + U_1^l$ holds is at most $\binom{q}{2}q_1$. By using the union bound over all those possible choices to obtain
  $$\Pr[\mathsf{bad}\gamma\text{-}\mathsf{coll}\text{-}1a] \le \frac{2q_1\binom{q}{2}}{N^2} \le \frac{q^2 q_1}{N^2}. \tag{4.10}$$
- badγ-coll-1b. $\exists i,j \in \mathcal{I}_{R*} \cap \mathcal{I}_{RR}$ and $i \ne j$ such that $S^i \ne S^j$ and $\widehat{S}^i = \widehat{S}^j$.
  In other words, $\exists i,j \in \mathcal{I}_{RR}$, such that $i \ne j \in \mathcal{I}_{\mathsf{dec}}$, and $k \in [i-1], l \in [j-1]$ such that
  $$R^i = R^k, R^j = R^l, \widehat{S}^i = \widehat{S}^j.$$
  Let's first fix the values for the indices i, j, k and l. The probability of the first two events $R^i = R^k$ and $R^j = R^l$ comes out to be $(1/N^2)$ due to the n-bit randomness over R^i and R^j. Moreover, the probability of the event $\widehat{S}^i = \widehat{S}^j$ comes out to be at most $2/N$ due to the randomness of $\widehat{S}^i$. However, the number of choices of indices (i,j,k,l) is at most q^4. By using the union bound over all those possible choices to obtain
  $$\Pr[\mathsf{bad}\gamma\text{-}\mathsf{coll}\text{-}1b] \le \frac{2q^4}{N^3}. \tag{4.11}$$

- badγ-coll-1c. $\exists i \in \mathcal{I}_{R*} \cap \mathcal{I}_R$ and $j \in \mathcal{I}_{R*} \cap \mathcal{I}_{RR}$ such that $S^i \neq S^j$ and $\widehat{S}^i = \widehat{S}^j$.

  In other words, $\exists i \in \mathcal{I}_R, j \in \mathcal{I}_{RR}$, such that $i \neq j$ and $j \in \mathcal{I}_{\mathsf{dec}}$, and $k \in [q_1], l \in [j-1]$ such that
  $$R^i + K_1 = U_1^k, R^j = R^l, \widehat{S}^i = \widehat{S}^j.$$
  Let's first fix the values for the indices i, j, k and l. The probability of the first two events $R^i + K_1 = U_1^k$ and $R^j = R^l$ comes out to be $(1/N^2)$ due to the n-bit randomness over k_1 and R^j. Moreover, the probability of the event $\widehat{S}^i = \widehat{S}^j$ comes out to be at most $2/N$ due to the randomness of $\widehat{S}^i$. However, the number of choices of indices (i, j, l) is at most q^3, and the number of choices for k is at most q_1. By using the union bound over all those possible choices to obtain
  $$\Pr[\text{bad}\gamma\text{-coll-1}c] \leq \frac{2q^3 q_1}{N^3}. \tag{4.12}$$

Adding the probabilities of the above three cases, we obtain
$$\Pr[\text{bad}\gamma\text{-coll-1}] \leq \frac{q^2 q_1}{N^2} + \frac{2q^4}{N^3} + \frac{2q^3 q_1}{N^3}. \tag{4.13}$$

Bounding badγ-coll-2

As before, we split the event into the following sub-cases and bound the probabilities of each of them.

- badγ-coll-2a. $\exists i, j \in \mathcal{I}_{S*} \cap \mathcal{I}_S$ and $i \neq j$ such that $R^i \neq R^j$ and $\widehat{R}^i = \widehat{R}^j$.

  In other words, $\exists i, j \in \mathcal{I}_S$, such that $i \neq j$, and $k, l \in [q_5]$ such that
  $$S^i + K_5 = U_5^k, S^j + K_5 = U_5^l, \widehat{R}^i = \widehat{R}^j.$$
  We can write the above event in an equivalent way as
  $$S^i + K_5 = U_5^k, S^i + S^j = U_5^k + U_5^l, \widehat{R}^i = \widehat{R}^j.$$
  Let's first fix the values for the indices i, j, k and l and without loss of generality, we assume that $i > j$. The probability of the event $S^i + K_5 = U_5^k$ comes out to be $(1/N)$ due to the n-bit randomness over the key K_5. Moreover, the probability of the event $\widehat{R}^i = \widehat{R}^j$ comes out to be at most $2/N$ due to the randomness of $\widehat{R}^i$. However, the number of choices of indices (i, j, k, l) such that $S^i + S^j = U_5^k + U_5^l$ holds is at most $\binom{q}{2} q_5$. By using the union bound over all those possible choices to obtain
  $$\Pr[\text{bad}\gamma\text{-coll-2}a] \leq \frac{2q_5 \binom{q}{2}}{N^2} \leq \frac{q^2 q_5}{N^2}. \tag{4.14}$$

- badγ-coll-2b. $\exists i, j \in \mathcal{I}_{S*} \cap \mathcal{I}_{SS}$ and $i \neq j$ such that $R^i \neq R^j$ and $\widehat{R}^i = \widehat{R}^j$. In other words, $\exists i, j \in \mathcal{I}_{SS}$, such that $i \neq j \in \mathcal{I}_{\mathsf{enc}}$, and $k \in [i-1], l \in [j-1]$ such that
  $$S^i = S^k, S^j = S^l, \widehat{R}^i = \widehat{R}^j.$$
  Let's first fix the values for the indices i, j, k and l. The probability of the first two events $S^i = S^k$ and $S^j = S^l$ comes out to be $(1/N^2)$ due to the n-bit randomness over S^i and S^j.

Moreover, the probability of the event $\widehat{R}^i = \widehat{R}^j$ comes out to be at most $2/N$ due to the randomness of $\widehat{R}^i$. However, the number of choices of indices (i, j, k, l) is at most q^4. By using the union bound over all those possible choices to obtain

$$\Pr[\mathsf{bad}\gamma\text{-}\mathsf{coll}\text{-}2b] \leq \frac{2q^4}{N^3}\,. \tag{4.15}$$

- badγ-coll-2c. $\exists i \in \mathcal{I}_{S*} \cap \mathcal{I}_S$ and $j \in \mathcal{I}_{S*} \cap \mathcal{I}_{SS}$ such that $R^i \neq R^j$ and $\widehat{R}^i = \widehat{R}^j$.

  In other words, $\exists i \in \mathcal{I}_S, j \in \mathcal{I}_{SS}$, such that $i \neq j$ and $j \in \mathcal{I}_{\mathsf{enc}}$, and $k \in [q_5], l \in [j-1]$ such that
  $$S^i + K_5 = U_5^k, S^j = S^l, \widehat{R}^i = \widehat{R}^j.$$
  Let's first fix the values for the indices i, j, k and l. The probability of the first two events $S^i + K_5 = U_5^k$ and $S^j = S^l$ comes out to be $(1/N^2)$ due to the n-bit randomness over K_5 and S^j. Moreover, the probability of the event $\widehat{R}^i = \widehat{R}^j$ comes out to be at most $2/N$ due to the randomness of $\widehat{R}^i$. However, the number of choices of indices (i, j, l) is at most q^3, and the number of choices for k is at most q_5. By using the union bound over all those possible choices to obtain

$$\Pr[\mathsf{bad}\gamma\text{-}\mathsf{coll}\text{-}2c] \leq \frac{2q^3 q_5}{N^3}\,. \tag{4.16}$$

Adding the probabilities of the above three cases, we obtain

$$\Pr[\mathsf{bad}\gamma\text{-}\mathsf{coll}\text{-}2] \leq \frac{q^2 q_5}{N^2} + \frac{2q^4}{N^3} + \frac{2q^3 q_5}{N^3}\,. \tag{4.17}$$

By combining Eqn. (4.13) and Eqn. (4.17), we have

$$\Pr[\mathsf{bad}\gamma\text{-}\mathsf{coll}] \leq \frac{q^2(q_1 + q_5)}{N^2} + \frac{4q^4}{N^3} + \frac{2q^3(q_1 + q_5)}{N^3}. \tag{4.18}$$

4.6.4 Bounding badγ-$\widehat{Y}$

Proposition 3. *Having defined the bad event badγ-$\widehat{Y}$ in Fig. 4.6, we have*

$$\Pr[\mathsf{bad}\gamma\text{-}\widehat{Y}] \leq \frac{4q^2(q_1 + q_5)}{N^2} + \frac{4q^3}{N^2}.$$

As before, to bound badγ-$\widehat{Y}$, we further split it into the following two cases:

- badγ-$\widehat{Y}$-1. $\exists i \in \mathcal{I}_*^c, j \in [q]$ and $i \neq j$ such that $R^i = R^j$ and $\widehat{S}^i + \widehat{S}^j = L^i + T^i + L^j + T^j$.
- badγ-$\widehat{Y}$-2. $\exists i \in \mathcal{I}_*^c, j \in [q]$ and $i \neq j$ such that $S^i = S^j$ and $\widehat{R}^i + \widehat{R}^j = L^i + T^i + L^j + T^j$.

Bounding badγ-$\widehat{Y}$-1

As before, we split the event into the following sub-cases and bound the probabilities of each of them.

- badγ-$\widehat{Y}$-1a $\exists i \in \mathcal{I}_R, j \in [q]$ and $i \neq j$ such that $R^i = R^j$ and $\widehat{S}^i + \widehat{S}^j = L^i + T^i + L^j + T^j$. In other words, $\exists i \in \mathcal{I}_R, j \in [q]$, with $i \neq j$ and $k \in [q_1]$ such that

  $$R^i + K_1 = U_1^k, R^i = R^j, \hat{S}^i + \hat{S}^j = L^i + T^i + L^j + T^j.$$

  Let's first fix the values for the indices i, j and k. The probability of the first event comes from the n-bit randomness over K_1 and the probability of the last event comes from the randomness over $\hat{S}^i$. Hence, the joint probability comes out to be at most $(2/N^2)$. However, the number of choices of indices i and j is at most $\binom{q}{2}$, and the number of choices for k is at most q_1. By using the union bound over all those possible choices to obtain

  $$\Pr[\text{bad}\gamma\text{-}\widehat{Y}\text{-}1a] \leq \frac{q^2 q_1}{N^2}\,. \tag{4.19}$$

- badγ-$\widehat{Y}$-1b. $\exists i \in \mathcal{I}_S, j \in [q]$ and $i \neq j$ such that $R^i = R^j$ and $\widehat{S}^i + \widehat{S}^j = L^i + T^i + L^j + T^j$. In other words, $\exists i \in \mathcal{I}_S, j \in [q]$, with $i \neq j$ and $k \in [q_5]$ such that

  $$S^i + K_5 = U_5^k, R^i = R^j, \hat{S}^i + \hat{S}^j = L^i + T^i + L^j + T^j.$$

  Now, we consider that $j \in \mathcal{I}_S$, as the analysis of this case is the involved one. Therefore, we have

  $$S^i + K_5 = U_5^k, S^j + K_5 = U_5^l, R^i = R^j, V_5^k + V_5^l = L^i + T^i + L^j + T^j, \tag{4.20}$$

  for some $l \in [q_5]$ and we equivalently write Eqn. (4.20) as

  $$S^i + K_5 = U_5^k, S^i + S^j = U_5^k + U_5^l, R^i = R^j, V_5^k + V_5^l = L^i + T^i + L^j + T^j. \tag{4.21}$$

  Now, we analyse this case in separate subcases:

  Case (a): We first assume the construction queries appear after the primitive queries and let $i < j$ and let j be an encryption query index (analysis for j to be a decryption query will be similar). Then, from the first equation, we use the randomness of K_5, and from the second equation, we use the randomness of S^j, which allows us to bound the probability of the event for a fixed choice of indices to at most $2/N^2$. Moreover, the number of tuples (i, j, k, l) such that Eqn. (4.21) holds is at most $\binom{q}{2}$ for choices of i and j, and the number of choices for k is at most q_5 which leaves a unique choice for l such that $V_5^k + V_5^l = L^i + T^i + L^j + T^j$ holds. Therefore, by varying all possible choices of indices, we bound the probability to at most $q^2 q_5/N^2$.

  Case (b): Now, we consider the case where the primitive queries appear after the construction queries and let $k < l$ and let l be a forward query index. Then from the first equation we use the randomness of K_5 and from the fourth equation, we use the randomness of V_5^l which allows us to bound the probability of the event for a fixed choice of indices, to at most $2/N^2$. Moreover, the number of tuples (i, j, k, l) such that Eqn. (4.21) holds is at most $\binom{q}{2}$ for choices of i and j, and the number of choices for k is at most q_5 which leaves a unique choice for l such that $S^i + S^j = U_5^k + U_5^l$ holds. Therefore, by varying all possible choices of indices, we bound the probability to at most $q^2 q_5/N^2$.

  Case (c): Similarly, if l is an inverse query index. Then from the first equation we use the randomness of K_5 and from the second equation, we use the randomness of U_5^l which allows us to bound the probability of the event for a fixed choice of indices, to at most

$2/N^2$. Moreover, the number of tuples (i, j, k, l) such that Eqn. (4.21) holds is at most $\binom{q}{2}$ for choices of i and j, and the number of choices for k is at most q_5 which leaves a unique choice for l such that $V_5^k + V_5^l = L^i + T^i + L^j + T^j$ holds. Therefore, by varying all possible choices of indices, we bound the probability to at most $q^2 q_5/N^2$.

By taking the union of all the above cases, we obtain

$$\Pr[\mathsf{bad}\gamma\text{-}\widehat{Y}\text{-}1b] \leq \frac{3q^2 q_5}{N^2}\,. \tag{4.22}$$

- $\mathsf{bad}\gamma$-$\widehat{Y}$-1*c*. $\exists i \in \mathcal{I}_{RR}, j \in [q]$ and $i \neq j$ such that $R^i = R^j$ and $\widehat{S}^i + \widehat{S}^j = L^i + T^i + L^j + T^j$. In other words, $\exists i \in \mathcal{I}_{RR}, j \in [q]$, with $i \neq j$ and $i \in \mathcal{I}_{\mathsf{dec}}$ and $k \in [i-1]$ such that

$$R^i = R^k, R^i = R^j, \hat{S}^i + \hat{S}^j = L^i + T^i + L^j + T^j.$$

Let's first fix the values for the indices i, j and k. The probability of the first event comes from the n-bit randomness over R^i and the probability of the last event comes from the randomness over $\hat{S}^i$. Hence, the joint probability comes out to be at most $(2/N^2)$. However, the number of choices of indices i and j is at most $\binom{q}{2}$, and the number of choices for k is at most q. By using the union bound over all those possible choices to obtain

$$\Pr[\mathsf{bad}\gamma\text{-}\widehat{Y}\text{-}1c] \leq \frac{q^3}{N^2}\,. \tag{4.23}$$

- $\mathsf{bad}\gamma$-$\widehat{Y}$-1*d*. $\exists i \in \mathcal{I}_{SS}, j \in [q]$ and $i \neq j$ such that $R^i = R^j$ and $\widehat{S}^i + \widehat{S}^j = L^i + T^i + L^j + T^j$. Analysis of this case is identical to the analysis of $\mathsf{bad}\gamma$-$\widehat{Y}$-1*c*., where we use the randomness of S^i as $i \in \mathcal{I}_{\mathsf{enc}}$. Hence, we obtain

$$\Pr[\mathsf{bad}\gamma\text{-}\widehat{Y}\text{-}1d] \leq \frac{q^3}{N^2}\,. \tag{4.24}$$

Adding the probabilities of the above four cases, we obtain

$$\Pr[\mathsf{bad}\gamma\text{-}\widehat{Y}\text{-}1] \leq \frac{q^2(q_1 + 3q_5)}{N^2} + \frac{2q^3}{N^2}\,. \tag{4.25}$$

Bounding $\mathsf{bad}\gamma$-$\widehat{Y}$-2

As before, we split the event into the following sub-cases and bound the probabilities of each of them.

- $\mathsf{bad}\gamma$-$\widehat{Y}$-2*a*. $\exists i \in \mathcal{I}_R, j \in [q]$ and $i \neq j$ such that $S^i = S^j$ and $\widehat{R}^i + \widehat{R}^j = L^i + T^i + L^j + T^j$. In other words, $\exists i \in \mathcal{I}_R, j \in [q]$, with $i \neq j$ and $k \in [q_1]$ such that

$$R^i + K_1 = U_1^k, S^i = S^j, \widehat{R}^i + \widehat{R}^j = L^i + T^i + L^j + T^j.$$

Now, we consider that $j \in \mathcal{I}_R$ as the analysis of this case is the involved one. Therefore, we have

$$R^i + K_1 = U_1^k, R^j + K_1 = U_1^l, S^i = S^j, V_1^k + V_1^l = L^i + T^i + L^j + T^j, \tag{4.26}$$

for some $l \in [q_1]$ and we equivalently write Eqn. (4.26) as

$$R^i + K_1 = U_1^k, R^i + R^j = U_1^k + U_1^l, S^i = S^j, V_1^k + V_1^l = L^i + T^i + L^j + T^j. \tag{4.27}$$

Now, we analyse this case in separate subcases:

Case (a): As before, we assume the construction queries appear after the primitive queries and let $i < j$ and let j be an encryption query index (analysis for j to be a decryption query will be similar). Then from the first equation we use the randomness of K_1 and from the third equation, we use the randomness of S^j which allows us to bound the probability of the event for a fixed choice of indices, to at most $2/N^2$. Moreover, the number of tuples (i, j, k, l) such that Eqn. (4.27) holds is at most $\binom{q}{2}$ for choices of i and j, and the number of choices for k is at most q_1 which leaves a unique choice for l such that $V_1^k + V_1^l = L^i + T^i + L^j + T^j$ holds. Therefore, by varying all possible choices of indices, we bound the probability to at most $q^2 q_1/N^2$.

Case (b): Analysis for this case is identical to the case (b) of bounding $\mathsf{bad}\gamma\text{-}\widehat{Y}\text{-}1c$. Therefore, by varying all possible choices of indices, we bound the probability to at most $q^2 q_1/N^2$.

Case (c): Analysis for this case is exactly identical to the case (c) of bounding $\mathsf{bad}\gamma\text{-}\widehat{Y}\text{-}1c$. Therefore, by varying all possible choices of indices, we bound the probability to at most $q^2 q_1/N^2$.

By taking the union of all the above cases, we obtain

$$\Pr[\mathsf{bad}\gamma\text{-}\widehat{Y}\text{-}2a] \leq \frac{3q^2 q_1}{N^2}. \tag{4.28}$$

- $\mathsf{bad}\gamma\text{-}\widehat{Y}\text{-}2b$. $\exists i \in \mathcal{I}_S, j \in [q]$ and $i \neq j$ such that $S^i = S^j$ and $\widehat{R}^i + \widehat{R}^j = L^i + T^i + L^j + T^j$. In other words, $\exists i \in \mathcal{I}_S, j \in [q]$, with $i \neq j$ and $k \in [q_5]$ such that

  $$S^i + K_5 = U_5^k, R^i = R^j, \widehat{R}^i + \widehat{R}^j = L^i + T^i + L^j + T^j.$$

  Let's first fix the values for the indices i, j and k. The probability of the first event comes from the n-bit randomness over K_5 and the probability of the last event comes from the randomness over $\widehat{R}^i$. Hence, the joint probability comes out to be at most $(2/N^2)$. However, the number of choices of indices i and j is at most $\binom{q}{2}$, and the number of choices for k is at most q_5. By using the union bound over all those possible choices to obtain

  $$\Pr[\mathsf{bad}\gamma\text{-}\widehat{Y}\text{-}2b] \leq \frac{q^2 q_5}{N^2}. \tag{4.29}$$

- $\mathsf{bad}\gamma\text{-}\widehat{Y}\text{-}2c$. $\exists i \in \mathcal{I}_{RR}, j \in [q]$ and $i \neq j$ such that $S^i = S^j$ and $\widehat{R}^i + \widehat{R}^j = L^i + T^i + L^j + T^j$. In other words, $\exists i \in \mathcal{I}_{RR}, j \in [q]$, with $i \neq j$ and $i \in \mathcal{I}_{\mathsf{dec}}$ and $k \in [i-1]$ such that

  $$R^i = R^k, S^i = S^j, \hat{R}^i + \hat{R}^j = L^i + T^i + L^j + T^j.$$

  Let's first fix the values for the indices i, j and k. The probability of the first event comes from the n-bit randomness over R^i and the probability of the last event comes from the randomness over $\hat{R}^i$. Hence, the joint probability comes out to be at most $(2/N^2)$. However, the number of choices of indices i and j is at most $\binom{q}{2}$, and the number of choices for k is at most q. By using the union bound over all those possible choices to obtain

  $$\Pr[\mathsf{bad}\gamma\text{-}\widehat{Y}\text{-}2c] \leq \frac{q^3}{N^2}. \tag{4.30}$$

- $\mathsf{bad}\gamma\text{-}\widehat{Y}\text{-}2d$. $\exists i \in \mathcal{I}_{SS}, j \in [q]$ and $i \neq j$ such that $S^i = S^j$ and $\widehat{R}^i + \widehat{R}^j = L^i + T^i + L^j + T^j$. Analysis of this case is identical to the analysis of $\mathsf{bad}\gamma\text{-}\widehat{Y}\text{-}2c.$, where we use the randomness of S^i as $i \in \mathcal{I}_{\mathsf{enc}}$. Hence, we obtain

$$\Pr[\mathsf{bad}\gamma\text{-}\widehat{Y}\text{-}2d] \leq \frac{q^3}{N^2}\,. \tag{4.31}$$

Adding the probabilities of the above four cases, we obtain

$$\Pr[\mathsf{bad}\gamma\text{-}\widehat{Y}\text{-}2] \leq \frac{q^2(3q_1+q_5)}{N^2} + \frac{2q^3}{N^2}\,. \tag{4.32}$$

By combining Eqn. (4.25) and Eqn. (4.32), we have

$$\Pr[\mathsf{bad}\gamma\text{-}\widehat{Y}] \leq \frac{4q^2(q_1+q_5)}{N^2} + \frac{4q^3}{N^2}. \tag{4.33}$$

4.7 Bounding the Ratio of Good Probabilities

Lemma 9. *Let $\eta = (\rho, \tau, \mathbf{K}, \gamma, \mu, \lambda)$ be any attainable transcript such that $\eta \notin \mathsf{bad}$. Suppose $q_1 + 2(\sqrt{q}+1) \leq q_2 + q_3 + q_4$, $q_5 + 2(\sqrt{q}+1) \leq q_2 + q_3 + q_4$ and $q + (q_1 + q_2 + \ldots + q_5) \leq N/2$. Then, we have*

$$\frac{\Pr[\mathcal{G}_1 \text{ yields } \eta]}{\Pr[\mathcal{G}_2 \text{ yields } \eta]} \geq 1 - \left(\frac{6q^3 + 4q^2(q_2+q_3+q_4) + 2qq_2q_3 + 2qq_2q_4 + 2qq_3q_4}{N^2} + \frac{8q^{3/2}}{N}\right).$$

Proof. Let $\eta = (\rho, \tau, \mathbf{K}, \gamma, \mu, \lambda)$ be a good transcript. We'll calculate the probability of obtaining η in the real world and an upper bound on its probability in the ideal world.

4.7.1 Real World

In the real world, there are N^5 choices for $\mathbf{K}$. Let Q_j denote the number of distinct queries to P_j for each $j \in [5]$. We first set aside the q_j primitive queries to P_j for each j, and hereafter count the additional distinct queries to each P_j that comes from the construction queries.

P_1 gets q_{R*} distinct queries in $\mathcal{I}_*$, and $q_R^{\mathcal{I}_{S*}}$ distinct queries in $\mathcal{I}_S$; and P_5 gets q_{S*} distinct queries in $\mathcal{I}_*$, and $q_S^{\mathcal{I}_{R*}}$ distinct queries in $\mathcal{I}_R$. Thus we have

$$Q_1 = q_1 + q_{R*} + q_R^{\mathcal{I}_{S*}}, \tag{4.34}$$
$$Q_5 = q_5 + q_{S*} + q_S^{\mathcal{I}_{R*}}. \tag{4.35}$$

For P_2, there are $q_X^{\mathcal{I}_R} + |\mathcal{I}_S|$ distinct queries in $\mathcal{I}_{\text{outer}}$, $|\mathcal{I}_{XX}|/2$ distinct queries in $\mathcal{I}_{XX}$, and $q_* - |\mathcal{I}_X| - |\mathcal{I}_{XX}|$ distinct queries in $\mathcal{I}_* \setminus (\mathcal{I}_X \cup \mathcal{I}_{XX})$, bringing the total to

$$\begin{aligned}
&q_X^{\mathcal{I}_R} + |\mathcal{I}_S| + |\mathcal{I}_{XX}|/2 + q_* - |\mathcal{I}_X| - |\mathcal{I}_{XX}| \\
&= q_X^{\mathcal{I}_R} + |\mathcal{I}_S| + q - |\mathcal{I}_R| - |\mathcal{I}_S| - |\mathcal{I}_X| - |\mathcal{I}_{XX}|/2 \\
&= q - |\mathcal{I}_X| - |\mathcal{I}_{XX}|/2 - |\mathcal{I}_R| + q_X^{\mathcal{I}_R}.
\end{aligned}$$

By a similar argument, we have $q - |\mathcal{I}_Z| - |\mathcal{I}_{ZZ}|/2 - |\mathcal{I}_S| + q_Z^{\mathcal{I}_S}$ distinct queries to P_4 in the construction queries. This gives us

$$Q_2 = q_2 + q - |\mathcal{I}_X| - |\mathcal{I}_{XX}|/2 - |\mathcal{I}_R| + q_X^{\mathcal{I}_R}, \tag{4.36}$$

$$Q_4 = q_4 + q - |\mathcal{I}_Z| - |\mathcal{I}_{ZZ}|/2 - |\mathcal{I}_S| + q_Z^{\mathcal{I}_S}. \tag{4.37}$$

Finally, we note that all queries to P_3 outside $\mathcal{I}_{\widehat{Y}} \cup \mathcal{I}_{\widehat{Y}\widehat{Y}}$ are distinct, and in addition there are $|\mathcal{I}_{\widehat{Y}\widehat{Y}}|/2$ distinct queries in $\mathcal{I}_{\widehat{Y}\widehat{Y}}$. This gives us

$$Q_3 = q_3 + q - |\mathcal{I}_{\widehat{Y}}| - |\mathcal{I}_{\widehat{Y}\widehat{Y}}|/2. \tag{4.38}$$

We have

$$\Pr[\mathcal{G}_1 \text{ yields } \eta] = \frac{1}{N^5} \cdot \frac{1}{(N)_{Q_1}} \cdot \frac{1}{(N)_{Q_2}} \cdot \frac{1}{(N)_{Q_3}} \cdot \frac{1}{(N)_{Q_4}} \cdot \frac{1}{(N)_{Q_5}}, \tag{4.39}$$

with $Q_1, \ldots, Q_5$ as in Eqns. (4.34)-(4.38). (We'll substitute the expressions later in Eqn. (4.39) when cancelling out the terms.)

4.7.2 Ideal World

In the ideal world, we first observe that ρ, τ, $\mathbf{K}$ are sampled independently of everything else, γ is sampled conditioned on $(\rho, \tau, \mathbf{K})$, and λ is sampled conditioned on $(\rho, \tau, \mathbf{K}, \gamma)$. This gives

$$\Pr[\mathcal{G}_2 \text{ yields } \eta] = \Pr_{\mathcal{O}_{\text{id}}}[\rho] \cdot \Pr_{\mathcal{O}_{\text{id}}}[\tau] \cdot \Pr_{\mathcal{O}_{\text{id}}}[\mathbf{K}] \cdot \Pr_{\mathcal{O}_{\text{id}}}[\gamma \mid \rho, \tau, \mathbf{K}] \cdot \Pr_{\mathcal{O}_{\text{id}}}[\lambda \mid \rho, \tau, \mathbf{K}, \gamma, \mu].. \tag{4.40}$$

Here $\Pr_{\mathcal{O}_{\text{id}}}[\theta]$ denoted the probability of an event θ occuring in the ideal world i.e in the game $\mathcal{G}_2$. Primitive queries are answered honestly, giving

$$\Pr_{\mathcal{O}_{\text{id}}}[\rho] = \frac{1}{(N)_{q_1}} \cdot \frac{1}{(N)_{q_2}} \cdot \frac{1}{(N)_{q_3}} \cdot \frac{1}{(N)_{q_4}} \cdot \frac{1}{(N)_{q_5}}. \tag{4.41}$$

Next, from Step-τa and Step-τb of the sampling, we get

$$\Pr_{\mathcal{O}_{\text{id}}}[\tau] = \frac{1}{N^{2q}}, \tag{4.42}$$

and from Step-K, we get

$$\Pr_{\mathcal{O}_{\text{id}}}[\mathbf{K}] = \frac{1}{N^5}. \tag{4.43}$$

A bound for γ.

We recall that the tricky part of sampling γ is how we sample it over $\mathcal{I}_*$. For each $d \in [q_*]$, we try to find an upper bound for the probability of sampling γ_*^d given $\gamma_*^{[d-1]}$ has already been sampled. We define

$$a_d := \min_{\gamma_*^{[d-1]}} \left| \Gamma_*^d \left[\gamma_*^{[d-1]} \right] \right|. \tag{4.44}$$

Then Step-γa gives

$$\Pr_{\mathcal{O}_{\text{id}}} \left[\gamma_*^d \mid \rho, \tau, \mathbf{K}, \gamma_*^{[d-1]} \right] \leq \frac{1}{a_d}. \tag{4.45}$$

Substituting Eqn. (4.44) in Eqn. (4.45) and taking the product over $d \in [q_*]$ gives

$$\Pr_{\mathcal{O}_{\text{id}}} \left[\gamma^{\mathcal{I}_*} \mid \rho, \tau, \mathbf{K} \right] = \Pr_{\mathcal{O}_{\text{id}}} \left[\gamma_*^{[q_*]} \mid \rho, \tau, \mathbf{K} \right] \leq \prod_{d=1}^{q_*} \frac{1}{a_d}. \tag{4.46}$$

This takes care of $\gamma^{\mathcal{I}_*}$. In $\mathcal{I}_{\text{outer}}$, $\textsf{Step-}\gamma\textsf{b}$ and $\textsf{Step-}\gamma\textsf{c}$ give

$$\Pr_{\mathcal{O}_{\text{id}}}\left[\gamma^{\mathcal{I}_{R*}\sqcup\mathcal{I}_{S*}} \mid \rho,\tau,\mathbf{K}\right] = \frac{1}{N^{q_S^{\mathcal{I}_{R*}}+q_R^{\mathcal{I}_{S*}}}}. \tag{4.47}$$

From Eqns. (4.46) and (4.47) we get

$$\Pr_{\mathcal{O}_{\text{id}}}[\gamma \mid \rho,\tau,\mathbf{K}] \leq \left(\prod_{d=1}^{q_*}\frac{1}{a_d}\right)\cdot\frac{1}{N^{q_S^{\mathcal{I}_{R*}}+q_R^{\mathcal{I}_{S*}}}}. \tag{4.48}$$

A bound for λ.

Again we recall that the tricky part of sampling λ is over $\mathcal{I}_{**}$. For each $h\in[q_{**}]$, we try to find an upper bound for the probability of sampling λ_{**}^h given $\lambda_{**}^{[h-1]}$ has already been sampled. We define

$$b_h := \min_{\lambda_{**}^{[h-1]}}\left|\Lambda_{**}^h\left[\lambda_{**}^{[h-1]}\right]\right|. \tag{4.49}$$

Then $\textsf{Step-}\lambda\textsf{a}$ gives

$$\Pr_{\mathcal{O}_{\text{id}}}\left[\lambda_{**}^h \mid \rho,\tau,\mathbf{K},\gamma,\mu,\lambda_{**}^{[h-1]}\right] \leq \frac{1}{b_h}. \tag{4.50}$$

From the definition of b_h and by taking the product of Eqn. (4.50) over $h\in[q_{**}]$ gives

$$\Pr_{\mathcal{O}_{\text{id}}}\left[\lambda^{\mathcal{I}_{**}} \mid \rho,\tau,\mathbf{K},\gamma,\mu\right] = \Pr_{\mathcal{O}_{\text{id}}}\left[\lambda_{**}^{[q_{**}]} \mid \rho,\tau,\mathbf{K},\gamma,\mu\right] \leq \prod_{h=1}^{q_{**}}\frac{1}{b_h}. \tag{4.51}$$

This takes care of $\lambda^{\mathcal{I}_{**}}$. On $\mathcal{I}_{\text{outer}}$ and $\mathcal{I}_{\text{inner}}$, from $\textsf{Step-}\lambda\textsf{b}$ we get

$$\Pr_{\mathcal{O}_{\text{id}}}\left[\lambda^{\mathcal{I}_R\sqcup\mathcal{I}_{XX}} \mid \rho,\tau,\mathbf{K},\gamma,\mu\right] = \frac{1}{N^{q_X^{\mathcal{I}_R}+|\mathcal{I}_{XX}|/2}}; \tag{4.52}$$

from $\textsf{Step-}\lambda\textsf{c}$ we get

$$\Pr_{\mathcal{O}_{\text{id}}}\left[\lambda^{\mathcal{I}_S\sqcup\mathcal{I}_{ZZ}} \mid \rho,\tau,\mathbf{K},\gamma,\mu\right] = \frac{1}{N^{q_Z^{\mathcal{I}_S}+|\mathcal{I}_{ZZ}|/2}}; \tag{4.53}$$

and finally, $\textsf{Step-}\lambda\textsf{d}$ and $\textsf{Step-}\lambda\textsf{e}$ give

$$\Pr_{\mathcal{O}_{\text{id}}}\left[\lambda^{\mathcal{I}_{\widehat{Y}\widehat{Y}}} \mid \rho,\tau,\mathbf{K},\gamma,\mu\right] = \frac{1}{N^{|\mathcal{I}_{RR}|+|\mathcal{I}_{SS}|}} \tag{4.54}$$

To keep the combined exponent of N readable, we'll use the notation

$$q^\dagger := q_X^{\mathcal{I}_R} + q_Z^{\mathcal{I}_S} + |\mathcal{I}_{RR}| + |\mathcal{I}_{SS}| + (|\mathcal{I}_{XX}| + |\mathcal{I}_{\widehat{Y}\widehat{Y}}| + |\mathcal{I}_{ZZ}|)/2. \tag{4.55}$$

Combining Eqns. (4.51), (4.52), (4.53), and (4.54) and substituting Eqn. (4.55) yields

$$\Pr_{\mathcal{O}_{\text{id}}}[\lambda \mid \rho,\tau,\mathbf{K},\gamma,\mu] \leq \left(\prod_{h=1}^{q_{**}}\frac{1}{b_h}\right)\cdot\frac{1}{N^{q^\dagger}}. \tag{4.56}$$

4.7.3 Bounding the ratio.

Plugging Eqns. (4.41), (4.42), (4.48), and (4.56) in Eqn. (4.40) gives

$$\Pr_{\mathcal{O}_{\mathrm{id}}}[\eta] \le \frac{1}{(N)_{q_1}} \cdot \frac{1}{(N)_{q_2}} \cdot \frac{1}{(N)_{q_3}} \cdot \frac{1}{(N)_{q_4}} \cdot \frac{1}{(N)_{q_5}} \cdot \frac{1}{N^5} \cdot \frac{1}{N^{2q}} \tag{4.57}$$

From Eqn. (4.39) and Eqn. (4.57), on writing $(N)_{Q_j}/(N)_{q_j}$ as $(N-q_j)_{Q_j-q_j}$ for each $j \in [5]$, we can calculate the H-ratio of η as

$$\begin{aligned} \mathsf{H}[\eta] &:= \frac{\Pr[\mathcal{G}_1 \text{ yields } = \eta]}{\Pr[\mathcal{G}_2 \text{ yields } = \eta]} \\ &\ge \frac{N^{q_S^{\mathcal{I}_{R*}} + q_R^{\mathcal{I}_{S*}}} \cdot \prod_{d=1}^{q_*} a_d}{(N-q_1)_{Q_1-q_1}(N-q_5)_{Q_5-q_5}} \\ &\qquad \cdot \frac{N^{2q} \cdot N^{q^\dagger} \cdot \prod_{h=1}^{q_{**}} b_h}{(N-q_2)_{Q_2-q_2}(N-q_3)_{Q_3-q_3}(N-q_4)_{Q_4-q_4}}. \end{aligned} \tag{4.58}$$

Note that we have

$$\begin{aligned} Q_2 - q_2 &= q - |\mathcal{I}_X| - |\mathcal{I}_{XX}|/2 - |\mathcal{I}_R| + q_X^{\mathcal{I}_R} \\ &= q_{**} + q_X^{\mathcal{I}_R} + |\mathcal{I}_{RR}| + |\mathcal{I}_S| + |\mathcal{I}_{SS}| \\ &\qquad + |\mathcal{I}_{XX}|/2 + |\mathcal{I}_{\widehat{Y}}| + |\mathcal{I}_{\widehat{Y}\widehat{Y}}| + |\mathcal{I}_Z| + |\mathcal{I}_{ZZ}|, \end{aligned} \tag{4.59}$$

so

$$\begin{aligned} (N-q_2)_{Q_2-q_2} \le (N-q_2)_{q_{**}} \cdot N^{q_X^{\mathcal{I}_R} + |\mathcal{I}_{XX}|/2} \\ \cdot N^{|\mathcal{I}_{RR}| + |\mathcal{I}_S| + |\mathcal{I}_{SS}| + |\mathcal{I}_{\widehat{Y}}| + |\mathcal{I}_{\widehat{Y}\widehat{Y}}| + |\mathcal{I}_Z| + |\mathcal{I}_{ZZ}|}. \end{aligned} \tag{4.60}$$

Similarly,

$$\begin{aligned} (N-q_3)_{Q_3-q_3} \le (N-q_3)_{q_{**}} \cdot N^{|\mathcal{I}_{RR}| + |\mathcal{I}_{SS}| + |\mathcal{I}_{\widehat{Y}\widehat{Y}}|/2} \\ \cdot N^{|\mathcal{I}_R| + |\mathcal{I}_S| + |\mathcal{I}_X| + |\mathcal{I}_{XX}| + |\mathcal{I}_Z| + |\mathcal{I}_{ZZ}|}, \end{aligned} \tag{4.61}$$

$$\begin{aligned} (N-q_4)_{Q_4-q_4} \le (N-q_4)_{q_{**}} \cdot N^{q_Z^{\mathcal{I}_S} + |\mathcal{I}_{ZZ}|/2} \\ \cdot N^{|\mathcal{I}_R| + |\mathcal{I}_{RR}| + |\mathcal{I}_{SS}| + |\mathcal{I}_Z| + |\mathcal{I}_{ZZ}| + |\mathcal{I}_{\widehat{Y}}| + |\mathcal{I}_{\widehat{Y}\widehat{Y}}|}. \end{aligned} \tag{4.62}$$

Multiplying (4.60), (4.61), and (4.62) gives

$$\begin{aligned} &(N-q_2)_{Q_2-q_2}(N-q_3)_{Q_3-q_3}(N-q_4)_{Q_4-q_4} \\ &\quad \le (N-q_2)_{q_{**}}(N-q_3)_{q_{**}}(N-q_4)_{q_{**}} \\ &\qquad \cdot N^{q_X^{\mathcal{I}_R} + q_Z^{\mathcal{I}_S} + |\mathcal{I}_{RR}| + |\mathcal{I}_{SS}| + (|\mathcal{I}_{XX}| + |\mathcal{I}_{\widehat{Y}\widehat{Y}}| + |\mathcal{I}_{ZZ}|)/2} \cdot N^{2q - 2q_{**}}. \end{aligned} \tag{4.63}$$

It follows that

$$\begin{aligned} &\frac{N^{2q} \cdot N^{q_X^{\mathcal{I}_R} + q_Z^{\mathcal{I}_S} + |\mathcal{I}_{RR}| + |\mathcal{I}_{SS}| + (|\mathcal{I}_{XX}| + |\mathcal{I}_{\widehat{Y}\widehat{Y}}| + |\mathcal{I}_{ZZ}|)/2}}{(N-q_2)_{Q_2-q_2}(N-q_3)_{Q_3-q_3}(N-q_4)_{Q_4-q_4}} \\ &\qquad\qquad \ge \frac{N^{2q_{**}}}{(N-q_2)_{q_{**}}(N-q_3)_{q_{**}}(N-q_4)_{q_{**}}}. \end{aligned} \tag{4.64}$$

Since $(N-q_1)_{Q_1-q_1} \leq (N-q_1)_{q_{R*}} \cdot N^{q_R^{\mathcal{I}_{S*}}}$ and $(N-q_5)_{Q_5-q_5} \leq (N-q_5)_{q_{S*}} \cdot N^{q_S^{\mathcal{I}_{R*}}}$, we also have

$$\frac{N^{q_S^{\mathcal{I}_{R*}}+q_R^{\mathcal{I}_{S*}}}}{(N-q_1)_{Q_1-q_1}(N-q_5)_{Q_5-q_5}} \geq \frac{1}{(N-q_1)_{q_{R*}}(N-q_5)_{q_{S*}}} \tag{4.65}$$

Substituting (4.64) and (4.65) in (4.58) gives

$$\mathsf{H}[\eta] \geq \frac{N^{2q_{**}} \prod_{h=1}^{q_{**}} b_h}{(N-q_2)_{q_{**}}(N-q_3)_{q_{**}}(N-q_4)_{q_{**}}} \cdot \frac{\prod_{d=1}^{q_*} a_d}{(N-q_1)_{q_{R*}}(N-q_5)_{q_{S*}}}. \tag{4.66}$$

We count $\prod_d a_d \cdot \prod_h b_h$ on each tree in sequence. Let $q^{(j)}$ be the number of queries in the j-th tree, and define $q_{R*}^{(j)} := |\{\ell \in [q_{R*}] \mid R_\ell \text{ is on the } j\text{-th tree}\}|$, $q_{S*}^{(j)} := |\{m \mid S_m \text{ is on the } j\text{-th tree}\}|$. Also, define the cumulative sums

$$q^{+(j)} := \sum_{l=1}^{j} q^{(l)}, \qquad q_{R*}^{+(j)} := \sum_{l=1}^{j} q_{R*}^{(l)}, \qquad q_{S*}^{+(j)} := \sum_{l=1}^{j} q_{S*}^{(l)}. \tag{4.67}$$

By our ordering, the queries in the j-th tree are precisely the ones with labels $d_1^{(j)} := q^{+(j-1)} + 1, \ldots, d_{q_{(j)}}^{(j)} := q^{+(j)}$.

Bounding a_d.

First, we consider the root node of the j-th tree. Here, both R and S are fresh, so we do not have to worry about $\mathsf{bad}\gamma\text{-}\widehat{Y}$. We just have to exclude the ranges of P_1 and P_5 sampled in primitive queries and earlier trees, giving

$$a_{d_1^{(j)}} \geq \left(N - q_1 - q_{R*}^{+(j-1)}\right) \cdot \left(N - q_5 - q_{S*}^{+(j-1)}\right). \tag{4.68}$$

For a query $d_k^{(j)}$, let $t^{d_k^{(j)}}$ be the number of elder siblings of its target node, plus the number of grandparents (0 for root or second-generation nodes and 1 for all subsequent nodes). Then, for an encryption query $d_k^{(j)}$, the number of earlier nodes with the same R (which can potentially give rise to $\mathsf{bad}\gamma\text{-}\widehat{Y}$) is exactly $t^{d_k^{(j)}}$, and the number of distinct $\widehat{S}$ already sampled before this node is $m^{d_k^{(j)}} - 1$. Thus we have

$$a_{d_k^{(j)}} \geq N - q_5 - \left(m^{d_k^{(j)}} - 1\right) - t^{d_k^{(j)}}, \tag{4.69}$$

Reasoning similarly for a decryption query $d_k^{(j)}$ we get

$$a_{d_k^{(j)}} \geq N - q_1 - \left(\ell^{d_k^{(j)}} - 1\right) - t^{d_k^{(j)}}. \tag{4.70}$$

We note that (4.69) and (4.70) do not depend on the tree except for the count t^d, and can simply be written as

$$a_d \geq N - q_5 - (m^d - 1) - t^d \tag{4.71}$$

and

$$a_d \geq N - q_1 - (\ell^d - 1) - t^d \tag{4.72}$$

for non-root encryption and decryption queries, respectively. Similarly, (4.68) can be written as

$$a_d \geq \big(N - q_1 - (\ell^d - 1)\big) \cdot \big(N - q_5 - (m^d - 1)\big) \tag{4.73}$$

for root queries, where $t^d = 0$. Let $t(\ell)$ (resp. $t(m)$) be defined as t^d where d is the first query (in the tree ordering) where R_ℓ (resp. S_m) appears. Then

$$\prod_{d=1}^{q_*} a_d \geq \prod_{\ell=1}^{q_{R*}} \left[N - q_1 - (\ell - 1) - t(\ell)\right] \cdot \prod_{m=1}^{q_{S*}} \left[N - q_5 - (m - 1) - t(m)\right]. \tag{4.74}$$

Bounding b_h.

For $h \in [q_{**}]$ let t^h_{**} be the number of elder siblings of its target node that come from $\mathcal{I}_{**}$, plus the number of grandparents that come from $\mathcal{I}_{**}$. While sampling λ^h_{**}, we need to maintain the three validity conditions on $\widehat{X}$, Y, and $\widehat{Z}$; since X, $\widehat{Y}$, and Z are all distinct on $\mathcal{I}_{**}$, we need to avoid collisions on $\widehat{X}$, Y, and $\widehat{Z}$ as well. For each of these three, in addition to the primitive queries, $h - 1$ distinct values have been sampled in the earlier nodes (in the tree-ordering), giving a total of $q_2 + q_3 + q_4 + 3(h - 1)$ candidates to avoid.

However, it turns out we can do slightly better. The key observation here is that for all earlier nodes with the same R or same S as this node, we avoid one of the three collisions for free! (For instance, $R^i = R^{i'}$ and $\widehat{X}^i \neq \widehat{X}^{i'}$ automatically imply that $Y^i = \widehat{X}^i + R^i \neq \widehat{X}^{i'} + R^{i'} = Y^{i'}$.) Thus, for the t^h_{**} earlier nodes with the same R or same S, we have one collision less to worry about. This shows that

$$b_h \geq N - (q_2 + q_3 + q_4) - 3(h - 1) + t^h_{**}. \tag{4.75}$$

Taking product over $[q_{**}]$ yields

$$\prod_{h=1}^{q_{**}} b_h \geq \prod_{h=1}^{q_{**}} \left[N - (q_2 + q_3 + q_4) - 3(h - 1) + t^h_{**}\right]. \tag{4.76}$$

This t^h_{**} term that we save here is crucial for the proof, as we use it to cancel out the corresponding $-t^d_*$ in the bound for a_d. That leaves us with reasonably simple bounds which we can approximate using standard techniques.

However, we still need to be careful, because $\mathcal{I}_{**}$ is slightly smaller than $\mathcal{I}_*$, which means that (i) each t^h_{**} will be slightly smaller than the corresponding t^d_*, and (ii) there will be slightly fewer t^h_{**} terms than $-t^d_*$ terms, leaving a few $-t^d_*$ terms that we can cancel out. Fortunately, the restrictions we have put on the bad events will be enough to bound these corner cases. We devote the rest of the section to deriving this concrete bound.

Completing the proof.

For $i \in \mathcal{I}_{**}$ (returning for the moment to the original query-order labelling), we look at $a_{d_i} b_{h_i}$. Suppose i is a non-root encryption query. Then from Eqns. (4.71) and (4.75) we get

$$a_{d_i} b_{h_i} \geq \left[N - q_5 - (m^{d_i} - 1) - t^{d_i}\right] \cdot \left[N - (q_2 + q_3 + q_4) - 3(h_i - 1) + t^{h_i}_{**}\right]. \tag{4.77}$$

We want to transfer the $t^{h_i}_{**}$ from the right parenthesis to the left. For any N_1, N_2, to claim $N_1(N_2 + t^{h_i}_{**}) \geq (N_1 + t^{h_i}_{**})N_2$, we just need to show that $N_1 \geq N_2$ (since $t^{h_i}_{**}$ is positive). Here we have $N_1 = N - [q_5 - (m^{d_i} - 1) - t^{d_i}]$ and $N_2 = N - [(q_2 + q_3 + q_4) + 3(h_i - 1)]$, so we just need to show that $(q_2+q_3+q_4)+3(h_i-1) > q_5-(m^{d_i}-1)-t^{d_i}$. Since $m^{d_i}\le d_i$, and $t^{d_i}\le d_i$, we get

$$\begin{aligned}
&q_2+q_3+q_4+3(h_i-1)-q_5-(m^{d_i}-1)-t^{d_i}\\
&\ge q_2+q_3+q_4+3h_i-3-q_5-d_i+1-d_i\\
&\ge q_2+q_3+q_4-2(d_i-h_i)-q_5-2\\
&\ge q_2+q_3+q_4-2|\mathcal{I}_{\text{inner}}|-q_5-2\\
&\ge q_2+q_3+q_4-(2\sqrt{q}+q_5+2)\ge 0, \qquad (4.78)
\end{aligned}$$

since $q_2+q_3+q_4\ge 2\sqrt{q}+q_5+2$. This allows us to carry out the intended transfer in (4.77) and get

$$\begin{aligned}
a_{d_i}b_{h_i} &\ge \left[N-q_5-(m^{d_i}-1)-(t^{d_i}-t^{h_i}_{**})\right]\cdot[N-(q_2+q_3+q_4)-3(h_i-1)]\\
&\ge \left[N-q_5-(m^{d_i}-1)-|\mathcal{I}_{\text{inner}}|\right]\cdot[N-(q_2+q_3+q_4)-3(h_i-1)]\\
&\ge \left[N-q_5-(m^{d_i}-1)-\sqrt{q}\right]\cdot[N-(q_2+q_3+q_4)-3(h_i-1)]\,. \qquad (4.79)
\end{aligned}$$

Similarly, when i is a non-root decryption query, we have

$$a_{d_i}b_{h_i}\ge\left[N-q_1-(\ell^{d_i}-1)-\sqrt{q}\right]\cdot[N-(q_2+q_3+q_4)-3(h_i-1)]\,. \qquad (4.80)$$

From here on, we can proceed to bound the two branches separately. For the parenthesis on the right of Eqn. (4.80), taking product over $\mathcal{I}_{**}$ gives

$$\prod_{i\in\mathcal{I}_{**}}[N-(q_2+q_3+q_4)-3(h_i-1)]=\prod_{h\in[q_{**}]}[N-(q_2+q_3+q_4)-3(h-1)]\,. \qquad (4.81)$$

We observe that

$$\begin{aligned}
&N^2\cdot(N-q_2-q_3-q_4-3(h-1))\\
&=(N-q_2-(h-1))(N-q_3-(h-1))(N-q_4-(h-1))\\
&\quad-N\left[(q_2+(h-1))(q_3+(h-1))+(q_2+(h-1))(q_4+(h-1))\right.\\
&\qquad\left.+(q_3+(h-1))(q_4+(h-1))\right]+(q_2+(h-1))(q_3+(h-1))(q_4+(h-1))\\
&\ge(N-q_2-(h-1))\cdot(N-q_3-(h-1))\cdot(N-q_4-(h-1))\\
&\quad\cdot\left[1-\frac{2}{N^2}\cdot[(q_2+(h-1))(q_3+(h-1))+(q_2+(h-1))(q_4+(h-1))\right.\\
&\qquad\qquad\left.+(q_3+(h-1))(q_4+(h-1))]\right]. \qquad (4.82)
\end{aligned}$$

Thus,

$$\begin{aligned}
&N^{2q_{**}}\cdot\left[\prod_{h=1}^{q_{**}}(N-(q_2+q_3+q_4)-3(h-1))\right]\\
&\qquad\ge(N-q_2)_{q_{**}}\cdot(N-q_3)_{q_{**}}\cdot(N-q_4)_{q_{**}}\cdot(1-\epsilon_0), \qquad (4.83)
\end{aligned}$$

where $\epsilon_0=2q[(q_2+q_{**})(q_3+q_{**})+(q_2+q_{**})(q_4+q_{**})+(q_3+q_{**})(q_4+q_{**})]/N^2$. This completes the bounding of the branch on the right of Eqn. (4.80). The final task that remains is to bound the branch on the left, combined with the a_d terms in $\mathcal{I}_{\text{inner}}$ (where the t_d did not get cancelled out). For each $i\in\mathcal{I}_*$, let w^i denote $\sqrt{q}$ if $i\in\mathcal{I}_{**}$ (corresponding to the $\sqrt{q}$ in the left parenthesis of Eqn. (4.80)) and q if $i \in \mathcal{I}_{\text{inner}}$ (corresponding to the $t(\ell)$ or $t(m)$ in Eqn. (4.74)). Let $w(\ell)$ (resp. $w(m)$) be defined as w^i where d_i is the first query where R_ℓ (resp. S_m) appears. Then

$$\begin{aligned}
&\prod_{\ell=1}^{q_{R*}} [N - q_1 - (\ell - 1) - w(\ell)] \cdot \prod_{m=1}^{q_{S*}} [N - q_5 - (m - 1) - w(m)] \\
&\geq (N - q_1)_{q_{R*}} (N - q_5)_{q_{S*}} \left[1 - \frac{2}{N} \cdot \left(\sum_{\ell=1}^{q_{R*}} w(\ell) + \sum_{m=1}^{q_{S*}} w(m) \right) \right] \\
&\geq (N - q_1)_{q_{R*}} (N - q_5)_{q_{S*}} \left[1 - \frac{4}{N} \cdot (\sqrt{q} \cdot |\mathcal{I}_{**}| + q \cdot |\mathcal{I}_{\text{inner}}|) \right] \\
&\geq (N - q_1)_{q_{R*}} (N - q_5)_{q_{S*}} \left(1 - \frac{8q^{3/2}}{N}\right). \qquad (4.84)
\end{aligned}$$

From Eqns. (4.79), (4.80), (4.83) and (4.84) we have

$$\begin{aligned}
\prod_{d=1}^{q_*} a_d \prod_{h=1}^{q_{**}} b_h \geq \frac{(N - q_2)_{q_{**}} (N - q_3)_{q_{**}} (N - q_4)_{q_{**}}}{N^{2q_{**}}} \\
\cdot (N - q_1)_{q_{R*}} (N - q_5)_{q_{S*}} \left(1 - \epsilon_0 - \frac{8q^{3/2}}{N}\right). \qquad (4.85)
\end{aligned}$$

Plugging in the value of ϵ_0 in Eqn. (4.85), using the inequality $q_{**} \leq q$ and substituting Eqn. (4.85) in Eqn. (4.66) gives

$$\mathsf{H}[\eta] \geq 1 - \left(\frac{6q^3 + 4q^2(q_2 + q_3 + q_4) + 2qq_2q_3 + 2qq_2q_4 + 2qq_3q_4}{N^2} + \frac{8q^{3/2}}{N} \right), \qquad (4.86)$$

which completes the proof.

□

4.8 Bounding the Probabilities of the Bad Events

4.8.1 Bounding badτ-switch

Let's first fix a pair of values for the indices i and j. If $j \in \mathcal{I}_{\mathsf{enc}}$, then the probability of the event $(S^j, T^j) = (S^i, T^i)$ comes out to be $(1/N) \cdot (1/N)$ due to the n-bit randomness over each of S^j and T^j. Similarly, if $j \in \mathcal{I}_{\mathsf{dec}}$, then the probability of the event $(L^j, R^j) = (L^i, R^i)$ comes out to be $(1/N) \cdot (1/N)$ due to the n-bit randomness over each of L^j and R^j. As we can choose the pair of indices (i, j) in $\binom{q}{2}$ ways, we use the union bound over all those possible choices to obtain

$$\Pr[\mathsf{bad}\tau\text{-}\mathsf{switch}] \leq \frac{\binom{q}{2}}{N^2}. \qquad (4.87)$$

4.8.2 Bounding badτ-$\widehat{Y}$

Let's first fix a pair of values for the indices i and j. If $j \in \mathcal{I}_{\mathsf{enc}}$, then the probability of each of the events $S^i = S^j$ and $L^i + T^i = L^j + T^j$ comes out to be $(1/N^2)$ due to the n- bit randomness over S^j and T^j respectively. Similarly, if $j \in \mathcal{I}_{\mathsf{dec}}$, then the probability of each of the events $R^i = R^j$ and $L^i + T^i = L^j + T^j$ comes out to be $(1/N^2)$ due to the n- bit randomness over R^j

and L^j respectively. As we can choose the pair of indices (i, j) in $\binom{q}{2}$ ways, we use the union bound over all those possible choices to obtain

$$\Pr[\mathsf{bad}\tau\text{-}\widehat{Y}] \leq \frac{\binom{q}{2}}{N^2}\,. \tag{4.88}$$

4.8.3 Bounding badτ-3path

Proposition 4. *Having defined the bad event badτ-3path in Fig. 4.3, we have*

$$\Pr[\textit{bad}\tau\text{-}\textit{3path}] \leq \frac{\binom{q}{3}}{N^2}.$$

To prove the proposition, let's first fix three distinct values for the indices i, j, and l. We'll study this bad event in the following four sub-cases.

- badτ-3path-1: If $j, l \in \mathcal{I}_{\mathsf{dec}}$, then $\Pr[R^i = R^j = R^l] = \Pr[R^i = R^j] \cdot \Pr[R^i = R^j = R^l | R^i = R^j]$ (as $\Pr[R^i = R^j = R^l | R^i \neq R^j] = 0$). This probability comes out to be $(1/N^2)$. The n-bit randomness for the first term on the RHS comes from R^j, and the same randomness for the second term on the RHS comes from R^l.
- badτ-3path-2: If $j, l \in \mathcal{I}_{\mathsf{enc}}$, then $\Pr[S^i = S^j = S^l] = \Pr[S^i = S^j] \cdot \Pr[S^i = S^j = S^l | S^i = S^j]$ (as $\Pr[S^i = S^j = S^l | S^i \neq S^j] = 0$). This probability comes out to be $(1/N^2)$. The n-bit randomness for the first term on the RHS comes from S^j and the same randomness for the second term on the RHS comes from S^l.
- badτ-3path-3: If $j \in \mathcal{I}_{\mathsf{dec}}$ and $l \in \mathcal{I}_{\mathsf{enc}}$, then the probability of each of the events $R^i = R^j = R^l$ and $S^i = S^j = S^l$ comes out to be $(1/N)$. The n-bit randomness comes from R^j and S^l, respectively.
- badτ-3path-4: If $j \in \mathcal{I}_{\mathsf{enc}}$ and $l \in \mathcal{I}_{\mathsf{dec}}$, then the probability of each of the events $R^i = R^j = R^l$ and $S^i = S^j = S^l$ comes out to be $(1/N)$. The n-bit randomness comes from R^l and S^j respectively.

As we can choose the 3-tuple of indices (i, j, l) in $\binom{q}{3}$ ways, we use the union bound over all those possible choices to obtain

$$\Pr[\mathsf{bad}\tau\text{-}\mathsf{3path}] \leq \frac{\binom{q}{3}}{N^2}\,. \tag{4.89}$$

4.8.4 Bounding badτ-3coll

Once we fix three distinct values for the indices i, j, and l, the analysis of this bad event exactly corresponds to the first two sub-cases of the previous bad event(e.g., badτ-3path). As we can choose the 3-tuple of indices (i, j, l) in $\binom{q}{3}$ ways, we use the union bound over all those possible choices to obtain

$$\Pr[\mathsf{bad}\tau\text{-}\mathsf{3coll}] \leq \frac{\binom{q}{3}}{N^2}\,. \tag{4.90}$$

4.8.5 Bounding badK-outer

Proposition 5. *Having defined the bad event badK-outer in Fig. 4.4, we have*

$$\Pr[\textit{badK-outer}] \le \frac{qq_1q_5 + q^2(q_1+q_5)}{N^2}.$$

To prove this proposition, we note that this bad event occurs when one of the following happens. Note that the event $\mathcal{I}_{RR} \cap \mathcal{I}_{SS} \neq \emptyset$ is an impossible event as $\mathcal{I}_{RR} \subseteq \mathcal{I}_{\mathsf{dec}}$ and $\mathcal{I}_{SS} \subseteq \mathcal{I}_{\mathsf{enc}}$ from definition.

- badK-outer-1 $\mathcal{I}_R \cap \mathcal{I}_S \neq \emptyset$. This bad event occurs when for some $i \in [q]$, $j \in [q_1]$ and $l \in [q_5]$, $R^i + K_1 = U_1^j$ and $S^i + K_5 = U_5^l$. Let's first fix the values for the indices i, j and l. Then the probability of each of the events $R^i + K_1 = U_1^j$ and $S^i + K_5 = U_5^l$ comes out to be $(1/N)$. The n-bit randomness comes from the keys K_1 and K_5, respectively. As we can choose the indices i, j and l in q, q_1 and q_5 ways respectively, we use the union bound over all those possible choices to obtain

$$\Pr[\mathcal{I}_R \cap \mathcal{I}_S \neq \emptyset] \le \frac{qq_1q_5}{N^2}\,. \tag{4.91}$$

- badK-outer-2 $\mathcal{I}_R \cap \mathcal{I}_{RR} \neq \emptyset$. This bad event occurs when for some $i \in \mathcal{I}_{\mathsf{dec}}$, $j \in [q_1]$ and $l \in [i-1]$, $R^i + K_1 = U_1^j$ and $R^i = R^l$. Let's first fix the values for the indices i, j and l. The probability of the event $R^i + K_1 = U_1^j$ comes out to be $(1/N)$. The n-bit randomness comes from the key K_1. The probability of the event $R^i = R^l$ also comes out to be $(1/N)$. The n-bit randomness comes from R^i as $i > l$ and $i \in \mathcal{I}_{\mathsf{dec}}$. As we can choose the pair of indices (i,l) in $\binom{q}{2}$ ways and the index j in q_1 ways, we use the union bound over all those possible choices to obtain

$$\Pr[\mathcal{I}_R \cap \mathcal{I}_{RR} \neq \emptyset] \le \frac{q_1\binom{q}{2}}{N^2}\,. \tag{4.92}$$

- badK-outer-3 $\mathcal{I}_S \cap \mathcal{I}_{SS} \neq \emptyset$. This bad event occurs when for some $i \in \mathcal{I}_{\mathsf{enc}}$, $j \in [q_5]$ and $l \in [i-1]$, $S^i + K_5 = U_5^j$ and $S^i = S^l$. Let's first fix the values for the indices i, j and l. The probability of the event $S^i + K_5 = U_5^j$ comes out to be $(1/N)$. The n-bit randomness comes from the key K_5. The probability of the event $S^i = S^l$ also comes out to be $(1/N)$. The n-bit randomness comes from S^i as $i > l$ and $i \in \mathcal{I}_{\mathsf{enc}}$. As we can choose the pair of indices (i,l) in $\binom{q}{2}$ ways and the index j in q_5 ways, we use the union bound over all those possible choices to obtain

$$\Pr[\mathcal{I}_S \cap \mathcal{I}_{SS} \neq \emptyset] \le \frac{q_5\binom{q}{2}}{N^2}\,. \tag{4.93}$$

- badK-outer-4 $\mathcal{I}_R \cap \mathcal{I}_{SS} \neq \emptyset$. This bad event occurs when for some $i \in \mathcal{I}_{\mathsf{enc}}$, $j \in [q_1]$ and $l \in [i-1]$, $R^i + K_1 = U_1^j$ and $S^i = S^l$. Let's first fix the values for the indices i, j and l. The probability of the event $R^i + K_1 = U_1^j$ comes out to be $(1/N)$. The n-bit randomness comes from the key K_1. The probability of the event $S^i = S^l$ also comes out to be $(1/N)$. The n-bit randomness comes from S^i as $i > l$ and $i \in \mathcal{I}_{\mathsf{enc}}$. As we can choose the pair of indices (i, l) in $\binom{q}{2}$ ways and the index j in q_1 ways, we use the union bound over all those possible choices to obtain

$$\Pr[\mathcal{I}_S \cap \mathcal{I}_{SS} \neq \emptyset] \leq \frac{q_1\binom{q}{2}}{N^2}\,. \tag{4.94}$$

- badK-outer-5 $\mathcal{I}_S \cap \mathcal{I}_{RR} \neq \emptyset$. This bad event occurs when for some $i \in \mathcal{I}_{\mathsf{dec}}$, $j \in [q_5]$ and $l \in [i-1]$, $S^i + K_5 = U_5^j$ and $R^i = R^l$. Let's first fix the values for the indices i, j and l. The probability of the event $S^i + K_5 = U_5^j$ comes out to be $(1/N)$. The n-bit randomness comes from the key K_5. The probability of the event $R^i = R^l$ also comes out to be $(1/N)$. The n-bit randomness comes from R^i as $i > l$ and $i \in \mathcal{I}_{\mathsf{dec}}$. As we can choose the pair of indices (i, l) in $\binom{q}{2}$ ways and the index j in q_5 ways, we use the union bound over all those possible choices to obtain

$$\Pr[\mathcal{I}_R \cap \mathcal{I}_{RR} \neq \emptyset] \leq \frac{q_5\binom{q}{2}}{N^2}\,. \tag{4.95}$$

Adding the probabilities of all these sub-cases, we obtain

$$\Pr[\mathsf{badK\text{-}outer}] \leq \frac{qq_1q_5 + q^2(q_1+q_5)}{N^2}\,. \tag{4.96}$$

4.8.6 Bounding badK-source

Proposition 6. *Having defined the bad event badK-source in Fig. 4.4, we have*

$$\Pr[\mathit{badK\text{-}source}] \leq \frac{(q_1+q_5)\binom{q}{2} + 2\binom{q}{3}}{N^2}.$$

This bad event occurs when one of the following happens.

- badK-source1. $\exists i \in \mathcal{I}_S$, $j \in \mathcal{I}_{RR}, i < j$ and $R^i = R^j$. In other words, $\exists i \in [q]$ and $j \in \mathcal{I}_{\mathsf{dec}}$ with $i < j$ and $l \in [q_5]$ such that $S^i + K_5 = U_5^l$ and $R^i = R^j$. Let's first fix the values for the indices i, j and l. The probability of each of the events $S^i + K_5 = U_5^l$ and $R^i = R^j$ comes out to be $(1/N)$. The n-bit randomness comes from the key K_5 and R_j, respectively. As we can choose the pair of indices (i, j) in $\binom{q}{2}$ ways and the index l in q_5 ways, we use the union bound over all those possible choices to obtain

$$\Pr[\mathsf{badK\text{-}source1}] \leq \frac{q_5\binom{q}{2}}{N^2}\,. \tag{4.97}$$

- badK-source2. $\exists i \in \mathcal{I}_{SS}, j \in \mathcal{I}_{RR}, i < j$ and $R^i = R^j$. In other words, $\exists l \in [q], i \in \mathcal{I}_{\mathsf{enc}}$ and $j \in \mathcal{I}_{\mathsf{dec}}$ with $k < i < j$ such that $R^i = R^j$ and $S^i = S^k$. Let's first fix the values for the indices i, j and l. The probability of each of the events $R^i = R^j$ and $S^i = S^k$ comes out to be $(1/N)$. The n-bit randomness comes from R_j and S_i respectively. As we can choose the 3-tuple of indices (i, j, l) in $\binom{q}{3}$ ways, we use the union bound over all those possible choices to obtain

$$\Pr[\mathsf{badK\text{-}source2}] \leq \frac{\binom{q}{3}}{N^2}\,. \tag{4.98}$$

- badK-source3. $\exists i \in \mathcal{I}_R, j \in \mathcal{I}_{SS}, i < j$ and $S^i = S^j$. In other words, $\exists i \in [q]$ and $j \in \mathcal{I}_{\mathsf{enc}}$ with $i < j$ and $l \in [q_1]$ such that $R^i + K_1 = U_1^l$ and $S^i = S^j$. Let's first fix the values for the indices i, j and l. The probability of each of the events $R^i + K_1 = U_1^l$ and $S^i = S^j$ comes out to be $(1/N)$. The n-bit randomness comes from the key K_1 and S_j, respectively. As we can choose the pair of indices (i, j) in $\binom{q}{2}$ ways and the index l in q_1 ways, we use the union bound over all those possible choices to obtain

$$\Pr[\text{badK-source3}] \leq \frac{q_1\binom{q}{2}}{N^2}\,. \tag{4.99}$$

- badK-source4. $\exists i \in \mathcal{I}_{RR}, j \in \mathcal{I}_{SS}, i < j$ and $S^i = S^j$. In other words, $\exists l \in [q], i \in \mathcal{I}_{\mathsf{dec}}$ and $j \in \mathcal{I}_{\mathsf{enc}}$ with $k < i < j$ such that $S^i = S^j$ and $R^i = R^k$. Let's first fix the values for the indices i, j and l. The probability of each of the events $S^i = S^j$ and $R^i = R^k$ comes out to be $(1/N)$. The n-bit randomness comes from S_j and R_i, respectively. As we can choose the 3-tuple of indices (i, j, l) in $\binom{q}{3}$ ways, we use the union bound over all those possible choices to obtain

$$\Pr[\text{badK-source4}] \leq \frac{\binom{q}{3}}{N^2}\,. \tag{4.100}$$

Adding the probabilities of all these sub-cases, we obtain

$$\Pr[\text{badK-source}] \leq \frac{(q_1 + q_5)\binom{q}{2} + 2\binom{q}{3}}{N^2}\,. \tag{4.101}$$

4.8.7 Bounding badμ-in&out

Proposition 7. *Having defined the bad event badμ-in&out in Fig. 4.7, we have*

$$\begin{aligned}\Pr[\textit{bad}\mu\textit{-in\&out}] \quad \leq \quad & \frac{q^2(3q_1 + 3q_5 + q_2 + q_3 + q_4)}{N^2} + \frac{5q^3}{N^2} + \frac{qq_1(q_3 + q_4 + q_5)}{N^2} \\ & + \frac{qq_5(q_2 + q_3 + q_4)}{N^2} + \frac{2q^2q_1q_5}{N^3} + \frac{2q^3(q_1 + q_5)}{N^3} + \frac{2q^2}{N^2}.\end{aligned}$$

This bad event occurs when $(\mathcal{I}_R \sqcup \mathcal{I}_S \sqcup \mathcal{I}_{RR} \sqcup \mathcal{I}_{SS}) \cap (\mathcal{I}_X \cup \mathcal{I}_{XX} \cup \mathcal{I}_{\widehat{Y}} \cup \mathcal{I}_{\widehat{Y}\widehat{Y}} \cup \mathcal{I}_Z \cup \mathcal{I}_{ZZ}) \neq \emptyset$. Note that by definition $\mathcal{I}_R \cap \mathcal{I}_{XX} = \emptyset$ and $\mathcal{I}_S \cap \mathcal{I}_{ZZ} = \emptyset$. We individually bound each of the bad events as follows:

- badμ-in&out-1. $\mathcal{I}_R \cap \mathcal{I}_X \neq \emptyset$. This bad event occurs when $\exists i \in [q], j \in [q_1]$ and $l \in [q_5]$ such that $R^i + K_1 = U_1^j$ and $X^i + K_2 = U_2^l$. Let's first fix the values for the indices i, j and l. The probability of each of the events $R^i + K_1 = U_1^j$ and $X^i + K_2 = U_2^l$ comes out to be $(1/N)$ due to the n-bit randomness over the keys K_1 and K_2 respectively. As we can choose the indices i, j and l in q, q_1 and q_5 ways respectively, we use the union bound over all those possible choices to obtain

$$\Pr[\text{bad}\mu\text{-in\&out-1}] \leq \frac{qq_1q_5}{N^2}\,. \tag{4.102}$$

- badμ-in&out-2. $\mathcal{I}_{RR} \cap \mathcal{I}_X \neq \emptyset$. This bad event occurs when $\exists i \in \mathcal{I}_{\mathsf{dec}}, j \in [i-1]$ and $l \in [q_2]$ such that $R^i = R^j$ and $X^i + K_2 = U_2^l$. Let's first fix the values for the indices i, j and l. The probability of each of the events $R^i = R^j$ and $X^j + K_2 = U_2^l$ comes out to be $(1/N)$ due to the n-bit randomness over R^i and K_2 respectively. As we can choose the pair of indices (i, j) in $\binom{q}{2}$ ways and the index l in q_2 ways, we use the union bound over all those possible choices to obtain

$$\Pr[\text{bad}\mu\text{-in\&out-2}] \leq \frac{q_2\binom{q}{2}}{N^2}\,. \tag{4.103}$$

- badμ-in&out-3. $\mathcal{I}_{RR} \cap \mathcal{I}_{XX} \neq \emptyset$. This bad event occurs when $\exists i \in \mathcal{I}_{\mathsf{dec}}, j \in [i-1]$, and $l \in [q]$ with $i \neq l$ such that $R^i = R^j$ and $X^i = X^l$, which we equivalently write as

$$R^i = R^j, \widehat{R}^i + \widehat{R}^l = L^i + L^l.$$

We analyse this event into two separate subcases: (a) when $l = j$ and if j is a decryption query, then the above event boils down to the event $R^i = R^j, L^i = L^j$, which triggers the bad event badτ-switch. Therefore, we analyse the case (b) when $l \neq j$. In this case, we use the randomness of R^i and $\widehat{R}^i$ to bound the above event to at most $(2/N^2)$ As we can choose the pair of indices $\{i, j\}$ in $\binom{q}{2}$ ways and for each of those choices, we can choose the index l in $(q-1)$ ways, we use the union bound over all those possible choices to obtain

$$\Pr[\text{bad}\mu\text{-in\&out-3}] \leq \frac{q^3}{N^2}\,. \tag{4.104}$$

- badμ-in&out-4. $\mathcal{I}_R \cap \mathcal{I}_{\widehat{Y}} \neq \emptyset$. This bad event occurs when $\exists i \in [q], j \in [q_1]$ and $k \in [q_3]$ such that $R^i + K_1 = U_1^j$ and $\widehat{Y}^i + K_3 = V_3^k$, which we equivalently write as

$$R^i + K_1 = U_1^j, \widehat{R}^i + L^i + \widehat{S}^i + T^i + K_3 = V_3^k.$$

For a fixed choice of indices, the probability of the event is at most $1/N^2$ due to the n-bit randomness over K_1 and K_3. We can choose the triplet of indices (i, j, k) in at most qq_1q_3 ways; we use the union bound over all those possible choices to obtain

$$\Pr[\text{bad}\mu\text{-in\&out-4}] \leq \frac{qq_1q_3}{N^2}\,. \tag{4.105}$$

- badμ-in&out-5. $\mathcal{I}_R \cap \mathcal{I}_{\widehat{Y}\widehat{Y}} \neq \emptyset$. This bad event occurs when $\exists i \in [q], j \in [q]$ and $k \in [q_1]$ such that $R^i + K_1 = U_1^k$ and $\widehat{Y}^i = \widehat{Y}^j$, which we equivalently write as

$$R^i + K_1 = U_1^k, \widehat{R}^i + \widehat{S}^i + \widehat{R}^j + \widehat{S}^j = L^i + L^j + T^i + T^j.$$

For a fixed choice of indices, the probability of the event is at most $2/N^2$ due to the n-bit randomness over K_1 and the n-bit randomness over $\widehat{S}^i$ (note that $i \notin \mathcal{I}_S$ and $i \notin \mathcal{I}_{SS}$). As we can choose the pair of indices $\{i, j\}$ in $\binom{q}{2}$ ways and for each of those choices, we can choose the index k in q_1 ways, we use the union bound over all those possible choices to obtain

$$\Pr[\text{bad}\mu\text{-in\&out-5}] \leq \frac{q^2q_1}{N^2}\,. \tag{4.106}$$

- badμ-in&out-6. $\mathcal{I}_R \cap \mathcal{I}_Z \neq \emptyset$. This bad event occurs when $\exists i \in [q], j \in [q_1]$ and $k \in [q_4]$ such that $R^i + K_1 = U_1^j$ and $Z^i + K_4 = U_4^k$, which we equivalently write as

$$R^i + K_1 = U_1^j, \widehat{S}^i + T^i + K_4 = U_4^k.$$

For a fixed choice of indices, the probability of the event is at most $1/N^2$ due to the n-bit randomness over K_1 and K_4. However, the total number of choices of the indices is at most qq_1q_4, we use the union bound over all those possible choices to obtain

$$\Pr[\text{bad}\mu\text{-in\&out-6}] \leq \frac{qq_1q_4}{N^2}\,. \tag{4.107}$$

- badμ-in&out-7. $\mathcal{I}_R \cap \mathcal{I}_{ZZ} \neq \emptyset$. This bad event occurs when $\exists i \in [q], j \in [q]$ and $k \in [q_1]$ such that $R^i + K_1 = U_1^k$ and $Z^i = Z^j$, which we equivalently write as

$$R^i + K_1 = U_1^k, \widehat{S}^i + T^i = \widehat{S}^j + T_j.$$

For a fixed choice of indices, the probability of the event is at most $2/N^2$ due to the n-bit randomness over K_1 and $\widehat{S}^i$ (note that $\widehat{S}^i$ is freshly sampled as $i \notin \mathcal{I}_S$ and $i \notin \mathcal{I}_{SS}$). However, the total number of choices of the indices is at most $\binom{q}{2}q_1$; we use the union bound over all those possible choices to obtain

$$\Pr[\text{bad}\mu\text{-in\&out-7}] \leq \frac{q^2q_1}{N^2}\,. \tag{4.108}$$

- badμ-in&out-8. $\mathcal{I}_S \cap \mathcal{I}_X \neq \emptyset$. Analysis of this case is similar to that of badμ-in&out-1., where we use the randomness of K_5 and K_2. Looking ahead, we bound the probability to be at most

$$\Pr[\text{bad}\mu\text{-in\&out-8}] \leq \frac{qq_2q_5}{N^2}\,. \tag{4.109}$$

- badμ-in&out-9. $\mathcal{I}_S \cap \mathcal{I}_{XX} \neq \emptyset$. Analysis of this case is again similar to that of badμ-in&out-7., where we use the randomness of K_5 and $\widehat{R}^i$. Looking ahead, we bound the probability to be at most

$$\Pr[\text{bad}\mu\text{-in\&out-9}] \leq \frac{q^2q_5}{N^2}\,. \tag{4.110}$$

- badμ-in&out-10. $\mathcal{I}_S \cap \mathcal{I}_{\widehat{Y}} \neq \emptyset$. Analysis of this case is again similar to that of badμ-in&out-4., where we use the randomness of K_5 and K_3. Looking ahead, we bound the probability to be at most

$$\Pr[\text{bad}\mu\text{-in\&out-10}] \leq \frac{qq_3q_5}{N^2}\,. \tag{4.111}$$

- badμ-in&out-11. $\mathcal{I}_S \cap \mathcal{I}_{\widehat{Y}\widehat{Y}} \neq \emptyset$. Analysis of this case is again similar to that of badμ-in&out-5., where we use the randomness of K_5 and $\widehat{R}^i$. Looking ahead, we bound the probability to be at most

$$\Pr[\text{bad}\mu\text{-in\&out-11}] \leq \frac{q^2q_5}{N^2}\,. \tag{4.112}$$

- badμ-in&out-12. $\mathcal{I}_S \cap \mathcal{I}_Z \neq \emptyset$. Analysis of this case is again similar to that of badμ-in&out-6., where we use the randomness of K_5 and K_4. Looking ahead, we bound the probability to be at most

$$\Pr[\text{bad}\mu\text{-in\&out-12}] \leq \frac{qq_4q_5}{N^2}\,. \tag{4.113}$$

- badμ-in&out-13. $\mathcal{I}_{RR} \cap \mathcal{I}_{\widehat{Y}} \neq \emptyset$. This bad event occurs when $\exists i \in \mathcal{I}_{\mathsf{dec}}, j \in [i-1]$ and $k \in [q_3]$ such that $R^i = R^j$ and $\widehat{Y}^i + K_3 = V_3^k$, which we equivalently write as

$$R^i = R^j, \widehat{R}^i + L^i + \widehat{S}^i + T^i + K_3 = V_3^k.$$

For a fixed choice of indices, the probability of the event is at most $1/N^2$ due to the n-bit randomness over R^i and K_3. We can choose the triplet of indices (i,j,k) is at most $\binom{q}{2}q_3$ ways, we use the union bound over all those possible choices to obtain

$$\Pr[\text{bad}\mu\text{-in\&out-13}] \leq \frac{q^2q_3}{2N^2}\,. \tag{4.114}$$

- badμ-in&out-14. $\mathcal{I}_{RR} \cap \mathcal{I}_{\widehat{Y}\widehat{Y}} \neq \emptyset$. This bad event occurs when $\exists i \in \mathcal{I}_{\mathsf{dec}}, j \in [i-1]$ and $k \in [q]$ such that $R^i = R^j$ and $\widehat{Y}^i = \widehat{Y}^k$, which we equivalently write as

$$R^i = R^j, \widehat{R}^i + \widehat{S}^i + \widehat{R}^k + \widehat{S}^k = L^i + L^k + T^i + T^k.$$

Now, we consider two separate subcases: (i) if $k = j$ and it is a decryption query, then the above event boils down to $R^i = R^j, L^i + L^j = T^i + T^j$ (assuming in both of the decryption queries S values are same). Then, using the randomness of R^i and L^i, we bound the above probability by at most $1/N^2$. Moreover, the number of choices for (i,j) to be at most $\binom{q}{2}$. Therefore, by using the union bound, the probability of the above event is at most $q^2/2N^2$.

Now, we consider the other case when $k \neq j$. In this case, we use the randomness of R^i and $\widehat{R}^i$ to bound the above event to at most $2/N^2$. The number of choices for triplets (i,j,k) is q^3. Therefore, by using the union bound, the probability of the above event is at most q^3/N^2.

Combining the above two cases, we obtain

$$\Pr[\text{bad}\mu\text{-in\&out-14}] \leq \frac{q^2}{2N^2} + \frac{q^3}{N^2}\,. \tag{4.115}$$

- badμ-in&out-15. $\mathcal{I}_{RR} \cap \mathcal{I}_Z \neq \emptyset$. This bad event occurs when $\exists i \in \mathcal{I}_{\mathsf{dec}}, j \in [i-1]$ and $k \in [q_4]$ such that $R^i = R^j$ and $Z^i + K_4 = U_4^k$, which we equivalently write as

$$R^i = R^j, \widehat{S}^i + T^i + K_4 = U_4^k.$$

For a fixed choice of indices, the probability of the event is at most $1/N^2$ due to the n-bit randomness over R^i and K_4. However, the total number of choices of the indices is at most $\binom{q}{2}q_4$, we use the union bound over all those possible choices to obtain

$$\Pr[\text{bad}\mu\text{-in\&out-15}] \leq \frac{q^2q_4}{2N^2}\,. \tag{4.116}$$

- badμ-in&out-16. $\mathcal{I}_{RR} \cap \mathcal{I}_{ZZ} \neq \emptyset$. This bad event occurs when $\exists i \in \mathcal{I}_{\mathsf{dec}}, j \in [i-1]$ and $k \in [q]$ such that $R^i = R^j$ and $Z^i = Z^k$, which we equivalently write as

$$R^i = R^j, \widehat{S}^i + T^i = \widehat{S}^k + T^k.$$

For a fixed choice of indices, the probability of the event is at most $2/N^2$ due to the n-bit randomness over $\widehat{R}^i$ and $\widehat{S}^i$ (note that $\widehat{S}^i$ is freshly sampled as $S^i \neq S^j$ and $i \notin \mathcal{I}_S$). However, the total number of choices of the indices is at most $\binom{q}{2}q$, we use the union bound over all those possible choices to obtain

$$\Pr[\text{bad}\mu\text{-in\&out-16}] \leq \frac{q^3}{2N^2}. \tag{4.117}$$

- badμ-in&out-17. $\mathcal{I}_{SS} \cap \mathcal{I}_X \neq \emptyset$. Analysis of this bad event is similar to that of badμ-in&out-12, where we use the randomness of S^i and K_2. Looking ahead, we bound the probability of the event to at most

$$\Pr[\text{bad}\mu\text{-in\&out-17}] \leq \frac{q_2\binom{q}{2}}{N^2}. \tag{4.118}$$

- badμ-in&out-18. $\mathcal{I}_{SS} \cap \mathcal{I}_{XX} \neq \emptyset$. This bad event occurs when $\exists i \in \mathcal{I}_{\mathsf{enc}}, j \in [i-1]$, and $l \in [q]$ with $i \neq l$ such that $S^i = S^j$ and $X^i = X^l$, which we equivalently write as

$$S^i = S^j, \widehat{R}^i + \widehat{R}^l = L^i + L^l.$$

We use the randomness of S^i and $\widehat{R}^i$ to bound the above event to at most $(2/N^2)$ As we can choose the pair of indices $\{i, j\}$ in $\binom{q}{2}$ ways and for each of those choices, we can choose the index l in $(q-1)$ ways, we use the union bound over all those possible choices to obtain

$$\Pr[\text{bad}\mu\text{-in\&out-18}] \leq \frac{q^3}{N^2}. \tag{4.119}$$

- badμ-in&out-19. $\mathcal{I}_{SS} \cap \mathcal{I}_{\widehat{Y}} \neq \emptyset$. Analysis of this bad event is similar to that of badμ-in&out-13, where we use the randomness of S^i and K_3. Looking ahead, we bound the probability of the event to at most

$$\Pr[\text{bad}\mu\text{-in\&out-19}] \leq \frac{q^2 q_3}{2N^2}. \tag{4.120}$$

- badμ-in&out-20. $\mathcal{I}_{SS} \cap \mathcal{I}_{\widehat{Y}\widehat{Y}} \neq \emptyset$. Analysis of this bad event is similar to that of badμ-in&out-16, where we use the randomness of S^i instead of R^i, wherever applicable. Looking ahead, we bound the probability of the above event to at most

$$\Pr[\text{bad}\mu\text{-in\&out-20}] \leq \frac{q^2}{2N^2} + \frac{q^3}{N^2}. \tag{4.121}$$

- badμ-in&out-21. $\mathcal{I}_{SS} \cap \mathcal{I}_Z \neq \emptyset$. Analysis of this bad event is similar to that of badμ-in&out-15, where we use the randomness of S^i and K_4. Looking ahead, we bound the above event to at most

$$\Pr[\text{bad}\mu\text{-in\&out-21}] \leq \frac{q^2 q_4}{2N^2}. \tag{4.122}$$

- badμ-in&out-22. $\mathcal{I}_{SS} \cap \mathcal{I}_{ZZ} \neq \emptyset$. Again, the analysis of this bad event is similar to that of badμ-in&out-3, where we use the randomness of S^i, wherever applicable. Looking ahead, we bound the above probability to be at most

$$\Pr[\text{bad}\mu\text{-in\&out-22}] \leq \frac{q^3}{2N^2}\,. \tag{4.123}$$

By combining Eqn. (4.102)-Eqn. (4.123), we obtain

$$\Pr[\text{bad}\mu\text{-in\&out}] \leq \frac{q^2(2q_1 + 2q_5 + q_2 + q_3 + q_4)}{N^2} + \frac{5q^3}{N^2} + \frac{qq_1(q_3 + q_4 + q_5)}{N^2} + \frac{qq_5(q_2 + q_3 + q_4)}{N^2} + \frac{2q^2}{N^2}. \tag{4.124}$$

4.8.8 Bounding badμ-source

Proposition 8. *Having defined the bad event badμ-source in Fig. 4.7, we have*

$$\Pr[\textit{bad}\mu\textit{-source}] \leq \frac{2\binom{q}{2}(q_1 + q_5)}{N^2}.$$

To prove the proposition, we first fix the values for the indices i, j and l.

- badμ-source-1. $i, j \in [q]$ with $i \neq j$ and $l \in [q_1]$ such that $R^i + K_1 = U_1^l$ and $\widehat{R}^i + \widehat{R}^j = L^i + L^j$. The probability of the event $R^i + K_1 = U_1^l$ comes out to be $(1/N)$ due to the randomness over the key K_1. The probability of the event $\widehat{R}^i + \widehat{R}^j = L^i + L^j$ comes out to be at most $(2/N)$ due to the randomness over $\widehat{R}^j$.

- badμ-source-2. $i, j \in [q]$ with $i \neq j$ and $l \in [q_5]$ such that $S^i + K_5 = U_5^l$ and $\widehat{S}^i + \widehat{S}^j = T^i + T^j$. The probability of the event $S^i + K_5 = U_5^l$ comes out to be $(1/N)$ due to the randomness over the key K_5. The probability of the event $\widehat{S}^i + \widehat{S}^j = T^i + T^j$ comes out to be at most $(2/N)$ due to the randomness over $\widehat{S}^j$.

As we can choose the pair of indices (i, j) in $2\binom{q}{2}$ ways and the index l in q_1 or q_5 ways (for badμ-source-1 and badμ-source-2 respectively), we use the union bound over all those possible choices to obtain

$$\Pr[\text{bad}\mu\text{-source}] \leq \frac{2\binom{q}{2}(q_1 + q_5)}{N^2}\,. \tag{4.125}$$

4.8.9 Bounding badμ-inner

Proposition 9. *Having defined the bad event badμ-inner in Fig. 4.7, we have*

$$\Pr[\textit{bad}\mu\textit{-inner}] \leq \frac{q(q_2q_3 + q_3q_4 + q_1q_4)}{N^2} + \frac{3q^2(q_2 + q_3 + q_4)}{N^2} + \frac{3q^3}{N^2}.$$

This bad event occurs when one of the following happens.

- badμ-inner-1. $\mathcal{I}_X \cap \mathcal{I}_{\widehat{Y}} \neq \emptyset$. This bad event occurs when $\exists i \in [q]$, $j \in [q_2]$ and $l \in [q_3]$ such that $X^i + K_2 = U_2^j$ and $\widehat{Y}^i + K_3 = V_3^l$. Let's first fix the values for the indices i, j and l. The probability of each of the events $X^i + K_2 = U_2^j$ and $\widehat{Y}^l = V_3^l$ comes out to be $(1/N)$

due to the randomness over the keys K_2 and K_3 respectively. As we can choose the indices i, j and l in q, q_2 and q_3 ways respectively, we use the union bound over all those possible choices to obtain

$$\Pr[\mathsf{bad}\mu\text{-}\mathsf{inner}\text{-}\mathsf{1}] \leq \frac{qq_2q_3}{N^2}\,. \tag{4.126}$$

- badμ-inner-2. $\mathcal{I}_{\widehat{Y}} \cap \mathcal{I}_Z \neq \emptyset$. This bad event occurs when $\exists i \in [q]$, $j \in [q_3]$ and $l \in [q_4]$ such that $\widehat{Y}^i + K_3 = V_3^j$ and $Z^i + K_4 = U_3^l$. Let's first fix the values for the indices i, j and l. The probability of each of the events $\widehat{Y}^i + K_3 = V_3^j$ and $Z^i + K_4 = U_3^l$ comes out to be $(1/N)$ due to the randomness over the keys K_3 and K_4 respectively. As we can choose the indices i, j and l in q, q_3 and q_4 ways respectively, we use the union bound over all those possible choices to obtain

$$\Pr[\mathsf{bad}\mu\text{-}\mathsf{inner}\text{-}\mathsf{2}] \leq \frac{qq_3q_4}{N^2}\,. \tag{4.127}$$

- badμ-inner-3. $\mathcal{I}_Z \cap \mathcal{I}_X \neq \emptyset$. This bad event occurs when $\exists i \in [q]$, $j \in [q_4]$ and $l \in [q_1]$ such that $Z^i + K_4 = U_4^j$ and $X^i + K_1 = U_1^l$. Let's first fix the values for the indices i, j and l. The probability of each of the events $Z^i + K_4 = U_4^j$ and $X^i + K_1 = U_1^l$ comes out to be $(1/N)$ due to the randomness over the keys K_4 and K_1 respectively. As we can choose the indices i, j and l in q, q_4 and q_1 ways respectively, we use the union bound over all those possible choices to obtain

$$\Pr[\mathsf{bad}\mu\text{-}\mathsf{inner}\text{-}\mathsf{3}] \leq \frac{qq_4q_1}{N^2}\,. \tag{4.128}$$

- badμ-inner-4. $\mathcal{I}_X \cap \mathcal{I}_{XX} \neq \emptyset$. This bad event occurs when $\exists i, j \in [q]$ with $i \neq j$ and $l \in [q_2]$ such that $X^i + K_2 = U_2^l$ and $X^i = X^j$. Let's first fix the values for the indices i, j and l. The probability of the event $X^i + K_2 = U_2^l$ comes out to be $(1/N)$ due to the randomness over the key K_2. The probability of the event $X^i = X^j$ comes out to be at most $(2/N)$ due to the n-bit randomness over X^i or X^j. As we can choose the pair of indices (i, j) in $2\binom{q}{2}$ and l in q_2 ways, we use the union bound over all those possible choices to obtain

$$\Pr[\mathsf{bad}\mu\text{-}\mathsf{inner}\text{-}\mathsf{4}] \leq \frac{2q_2\binom{q}{2}}{N^2}\,. \tag{4.129}$$

- badμ-inner-5. $\mathcal{I}_X \cap \mathcal{I}_{\widehat{Y}\widehat{Y}} \neq \emptyset$. This bad event occurs when $\exists i, j \in [q]$ with $i \neq j$ and $l \in [q_2]$ such that $X^i + K_2 = U_2^l$ and $\widehat{Y}^i = \widehat{Y}^j$. Let's first fix the values for the indices i, j and l. The probability of the event $X^i + K_2 = U_2^l$ comes out to be $(1/N)$ due to the randomness over the key K_2. The probability of the event $\widehat{Y}^i = \widehat{Y}^j$ comes out to be at most $(2/N)$ due to the n-bit randomness over $\widehat{Y}^i$ or $\widehat{Y}^j$. As we can choose the pair of indices (i, j) in $2\binom{q}{2}$ and l in q_2 ways, we use the union bound over all those possible choices to obtain

$$\Pr[\mathsf{bad}\mu\text{-}\mathsf{inner}\text{-}\mathsf{5}] \leq \frac{2q_2\binom{q}{2}}{N^2}\,. \tag{4.130}$$

- badμ-inner-6. $\mathcal{I}_X \cap \mathcal{I}_{ZZ} \neq \emptyset$. This bad event occurs when $\exists i, j \in [q]$ with $i \neq j$ and $l \in [q_2]$ such that $X^i + K_2 = U_2^l$ and $Z^i = Z^j$. Let's first fix the values for the indices i, j and l. The probability of the event $X^i + K_2 = U_2^l$ comes out to be $(1/N)$ due to the randomness over the key K_2. The probability of the event $Z^i = Z^j$ comes out to be at most $(2/N)$ due to the n-bit randomness over Z^i or Z^j. As we can choose the pair of indices (i, j) in $2\binom{q}{2}$ and l in q_2 ways, we use the union bound over all those possible choices to obtain

$$\Pr[\text{bad}\mu\text{-inner-6}] \leq \frac{2q_2\binom{q}{2}}{N^2} . \tag{4.131}$$

- badμ-inner-7. $\mathcal{I}_{\widehat{Y}} \cap \mathcal{I}_{XX} \neq \emptyset$. This bad event occurs when $\exists i, j \in [q]$ with $i \neq j$ and $l \in [q_3]$ such that $\widehat{Y}^i + K_3 = U_3^l$ and $X^i = X^j$. Let's first fix the values for the indices i, j and l. The probability of the event $\widehat{Y}^i + K_3 = U_3^l$ comes out to be $(1/N)$ due to the randomness over the key K_3. The probability of the event $X^i = X^j$ comes out to be at most $(2/N)$ due to the n-bit randomness over X^i or X^j. As we can choose the pair of indices (i, j) in $2\binom{q}{2}$ and l in q_3 ways, we use the union bound over all those possible choices to obtain

$$\Pr[\text{bad}\mu\text{-inner-7}] \leq \frac{2q_3\binom{q}{2}}{N^2} . \tag{4.132}$$

- badμ-inner-8. $\mathcal{I}_{\widehat{Y}} \cap \mathcal{I}_{\widehat{Y}\widehat{Y}} \neq \emptyset$. This bad event occurs when $\exists i, j \in [q]$ with $i \neq j$ and $l \in [q_3]$ such that $\widehat{Y}^i + K_3 = U_3^l$ and $\widehat{Y}^i = \widehat{Y}^j$. Let's first fix the values for the indices i, j and l. The probability of the event $\widehat{Y}^i + K_3 = U_3^l$ comes out to be $(1/N)$ due to the randomness over the key K_3. The probability of the event $\widehat{Y}^i = \widehat{Y}^j$ comes out to be at most $(2/N)$ due to the n-bit randomness over $\widehat{Y}^i$ or $\widehat{Y}^j$. As we can choose the pair of indices (i, j) in $2\binom{q}{2}$ and l in q_3 ways, we use the union bound over all those possible choices to obtain

$$\Pr[\text{bad}\mu\text{-inner-8}] \leq \frac{2q_3\binom{q}{2}}{N^2} . \tag{4.133}$$

- badμ-inner-9. $\mathcal{I}_{\widehat{Y}} \cap \mathcal{I}_{ZZ} \neq \emptyset$. This bad event occurs when $\exists i, j \in [q]$ with $i \neq j$ and $l \in [q_3]$ such that $\widehat{Y}^i + K_3 = U_3^l$ and $Z^i = Z^j$. Let's first fix the values for the indices i, j and l. The probability of the event $\widehat{Y}^i + K_3 = U_3^l$ comes out to be $(1/N)$ due to the randomness over the key K_3. The probability of the event $Z^i = Z^j$ comes out to be at most $(2/N)$ due to the n-bit randomness over Z^i or Z^j. As we can choose the pair of indices (i, j) in $2\binom{q}{2}$ and l in q_3 ways, we use the union bound over all those possible choices to obtain

$$\Pr[\text{bad}\mu\text{-inner-9}] \leq \frac{2q_3\binom{q}{2}}{N^2} . \tag{4.134}$$

- badμ-inner-10. $\mathcal{I}_Z \cap \mathcal{I}_{XX} \neq \emptyset$. This bad event occurs when $\exists i, j \in [q]$ with $i \neq j$ and $l \in [q_4]$ such that $Z^i + K_4 = U_4^l$ and $X^i = X^j$. Let's first fix the values for the indices i, j and l. The probability of the event $Z^i + K_4 = U_4^l$ comes out to be $(1/N)$ due to the randomness over the key K_4. The probability of the event $X^i = X^j$ comes out to be at most $(2/N)$ due to the n-bit randomness over X^i or X^j. As we can choose the pair of indices (i, j) in $2\binom{q}{2}$ and l in q_4 ways, we use the union bound over all those possible choices to obtain

$$\Pr[\text{bad}\mu\text{-inner-10}] \leq \frac{2q_4\binom{q}{2}}{N^2} . \tag{4.135}$$

- badμ-inner-11. $\mathcal{I}_Z \cap \mathcal{I}_{\widehat{Y}\widehat{Y}} \neq \emptyset$. This bad event occurs when $\exists i, j \in [q]$ with $i \neq j$ and $l \in [q_4]$ such that $Z^i + K_4 = U_4^l$ and $\widehat{Y}^i = \widehat{Y}^j$. Let's first fix the values for the indices i, j and l. The probability of the event $Z^i + K_4 = U_4^l$ comes out to be $(1/N)$ due to the randomness over the key K_4. The probability of the event $\widehat{Y}^i = \widehat{Y}^j$ comes out to be at most $(2/N)$ due to the n-bit randomness over $\widehat{Y}^i$ or $\widehat{Y}^j$. As we can choose the pair of indices (i,j) in $2\binom{q}{2}$ and l in q_4 ways, we use the union bound over all those possible choices to obtain

$$\Pr[\mathsf{bad}\mu\text{-}\mathsf{inner}\text{-}\mathsf{11}] \leq \frac{2q_4\binom{q}{2}}{N^2}\,. \tag{4.136}$$

- badμ-inner-12. $\mathcal{I}_Z \cap \mathcal{I}_{ZZ} \neq \emptyset$. This bad event occurs when $\exists i,j \in [q]$ with $i \neq j$ and $l \in [q_4]$ such that $Z^i + K_4 = U_4^l$ and $Z^i = Z^j$. Let's first fix the values for the indices i, j and l. The probability of the event $Z^i + K_4 = U_4^l$ comes out to be $(1/N)$ due to the randomness over the key K_4. The probability of the event $Z^i = Z^j$ comes out to be at most $(2/N)$ due to the n-bit randomness over Z^i or Z^j. As we can choose the pair of indices (i,j) in $2\binom{q}{2}$ and l in q_4 ways, we use the union bound over all those possible choices to obtain

$$\Pr[\mathsf{bad}\mu\text{-}\mathsf{inner}\text{-}\mathsf{12}] \leq \frac{2q_4\binom{q}{2}}{N^2}\,. \tag{4.137}$$

- badμ-inner-13. $\mathcal{I}_{XX} \cap \mathcal{I}_{\widehat{Y}\widehat{Y}} \neq \emptyset$. This bad event occurs when $\exists i,j,l \in [q]$ with $i \neq j$ and $i \neq l$ such that $X^i = X^j$ and $\widehat{Y}^i = \widehat{Y}^l$. Let's first fix the values for the indices i, j and l. The probability of each of the events comes out to be at most $(2/N)$ due to the n-bit randomness of X^i or X^j and $\widehat{Y}^i$ or $\widehat{Y}^j$. As we can choose the index i in q ways and for each of those choices, we can choose each of the indices j and l in $(q-1)$ ways, we use the union bound over all those possible choices to obtain

$$\Pr[\mathsf{bad}\mu\text{-}\mathsf{inner}\text{-}\mathsf{13}] \leq \frac{q(q-1)^2}{N^2}\,. \tag{4.138}$$

- badμ-inner-14. $\mathcal{I}_{\widehat{Y}\widehat{Y}} \cap \mathcal{I}_{ZZ} \neq \emptyset$. This bad event occurs when $\exists i,j,l \in [q]$ with $i \neq j$ and $i \neq l$ such that $\widehat{Y}^i = \widehat{Y}^j$ and $Z^i = Z^l$. Let's first fix the values for the indices i, j and l. The probability of each of the events comes out to be at most $(2/N)$ due to the n-bit randomness of $\widehat{Y}^i$ or $\widehat{Y}^j$ and Z^i or Z^j. As we can choose the index i in q ways and for each of those choices, we can choose each of the indices j and l in $(q-1)$ ways, we use the union bound over all those possible choices to obtain

$$\Pr[\mathsf{bad}\mu\text{-}\mathsf{inner}\text{-}\mathsf{14}] \leq \frac{q(q-1)^2}{N^2}\,. \tag{4.139}$$

- badμ-inner-15. $\mathcal{I}_{ZZ} \cap \mathcal{I}_{XX} \neq \emptyset$. This bad event occurs when $\exists i,j,l \in [q]$ with $i \neq j$ and $i \neq l$ such that $Z^i = Z^j$ and $X^i = X^l$. Let's first fix the values for the indices i, j and l. The probability of each of the events comes out to be at most $(2/N)$ due to the n-bit randomness of Z^i or Z^j and X^i or X^j. As we can choose the index i in q ways and for each of those choices, we can choose each of the indices j and l in $(q-1)$ ways, we use the union bound over all those possible choices to obtain

$$\Pr[\mathsf{bad}\mu\text{-}\mathsf{inner}\text{-}\mathsf{15}] \leq \frac{q(q-1)^2}{N^2}\,. \tag{4.140}$$

By combining Eqn. (4.126)-Eqn. (4.140), we have

$$\Pr[\mathsf{bad}\mu\text{-}\mathsf{inner}] \leq \frac{q(q_2q_3 + q_3q_4 + q_1q_4)}{N^2} + \frac{3q^2(q_2+q_3+q_4)}{N^2} + \frac{3q^3}{N^2}. \tag{4.141}$$

4.8.10 Bounding badμ-3coll

Proposition 10. *Having defined the bad event badμ-3coll in Fig. 4.7, we have*

$$\Pr[\text{bad}\mu\text{-3coll}] \leq \frac{4\binom{q}{3}}{N^2}.$$

To prove the proposition, we first fix the values for the indices i, j and l.

- badμ-3coll-1. $i, j, l \in [q]$ with $i < j < l$ such that $X^i = X^j = X^l$. We can write $\Pr[X^i = X^j = X^l] = \Pr[X^i = X^j] \cdot \Pr[X^i = X^j = X^l | X^i = X^j]$ (as $\Pr[X^i = X^j = X^l | X^i \neq X^j] = 0$). Each term on the RHS can be at most $(2/N)$ due to the randomness over X^j and X^l, respectively.

- badμ-3coll-2. $i, j, l \in [q]$ with $i < j < l$ such that $\widehat{Y}^i = \widehat{Y}^j = \widehat{Y}^l$. We can write $\Pr[\widehat{Y}^i = \widehat{Y}^j = \widehat{Y}^l] = \Pr[\widehat{Y}^i = \widehat{Y}^j] \cdot \Pr[\widehat{Y}^i = \widehat{Y}^j = \widehat{Y}^l | \widehat{Y}^i = \widehat{Y}^j]$ (as $\Pr[\widehat{Y}^i = \widehat{Y}^j = \widehat{Y}^l | \widehat{Y}^i \neq \widehat{Y}^j] = 0$). Each term on the RHS can be at most $(2/N)$ due to the randomness over $\widehat{Y}^j$ and $\widehat{Y}^l$ respectively.

- badμ-3coll-3. $i, j, l \in [q]$ with $i < j < l$ such that $Z^i = Z^j = Z^l$. We can write $\Pr[Z^i = Z^j = Z^l] = \Pr[Z^i = Z^j] \cdot \Pr[Z^i = Z^j = Z^l | Z^i = Z^j]$ (as $\Pr[Z^i = Z^j = Z^l | Z^i \neq Z^j] = 0$). Each term on the RHS can be at most $(2/N)$ due to the randomness over Z^j and Z^l respectively.

As we can choose the 3-tuple of indices (i, j, l) in $\binom{q}{3}$ ways, we use the union bound over all those possible choices to obtain

$$\Pr[\text{bad}\mu\text{-3col}] \leq \frac{4\binom{q}{3}}{N^2}. \tag{4.142}$$

4.8.11 Bounding badμ-size

Proposition 11. *Having defined the bad event badμ-size in Fig. 4.7, we have*

$$\Pr[\text{bad}\mu\text{-size}] \leq \frac{q^{1/2}(q_2 + q_3 + q_4)}{N} + \frac{2q^{3/2}}{N}.$$

We say that the bad event badμ-size happens if one of the following events happens.

- badμ-size-prim This event holds if either of the following three events holds:
  - badμ-size-$\mathcal{I}_X$: This event holds if $|\mathcal{I}_X| > q^{1/2}$.
  - badμ-size-$\mathcal{I}_{\widehat{Y}}$: This event holds if $|\mathcal{I}_{\widehat{Y}}| > q^{1/2}$.
  - badμ-size-$\mathcal{I}_Z$: This event holds if $|\mathcal{I}_Z| > q^{1/2}$.

- badμ-size-coll This event holds if either of the following three events holds:
  - badμ-size-$\mathcal{I}_{XX}$: This event holds if $|\mathcal{I}_{XX}| > q^{1/2}$.
  - badμ-size-$\mathcal{I}_{\widehat{Y}\widehat{Y}}$: This event holds if $|\mathcal{I}_{\widehat{Y}\widehat{Y}}| > q^{1/2}$.
  - badμ-size-$\mathcal{I}_{ZZ}$: This event holds if $|\mathcal{I}_{ZZ}| > q^{1/2}$.

Bounding badμ-size-prim

To bound this event, we bound each of the following events: badμ-size-$\mathcal{I}_X$, badμ-size-$\mathcal{I}_{\widehat{Y}}$, and bad$\mu$-size-$\mathcal{I}_Z$. We begin with bounding the size of $|\mathcal{I}_X|$. Let for each $i \in [q]$, $\mathbb{I}_i$ be an indicator random variable that takes the value 1 if there exists a $j \in [q_2]$ such that $X^i + K_2 = U_2^j$. Note that the probability of this event holds is at most q_2/N using the randomness of key K_2, i.e., for a fixed $i \in [q]$,

$$\Pr[\mathbb{I}_i = 1] \leq \frac{q_2}{N}.$$

Therefore, by the linearity of expectations and by applying Markov's inequality, we have

$$\Pr[|\mathcal{I}_X| > q^{1/2}] \leq \frac{q^{1/2} q_2}{N} \approx \frac{q^{3/2}}{N}, \qquad \text{(provided, } q_2 \approx q\text{)}.$$

Similarly, we can show that

$$\Pr[|\mathcal{I}_{\widehat{Y}}| > q^{1/2}] \leq \frac{q^{1/2} q_3}{N}, \quad \Pr[|\mathcal{I}_Z| > q^{1/2}] \leq \frac{q^{1/2} q_4}{N}.$$

By combining the above three cases, we have

$$\Pr[\text{bad}\mu\text{-size-prim}] \leq \frac{q^{1/2}(q_2 + q_3 + q_4)}{N}. \tag{4.143}$$

Bounding badμ-size-coll

To bound this event, we bound each of the following events: badμ-size-$\mathcal{I}_{XX}$, badμ-size-$\mathcal{I}_{\widehat{Y}\widehat{Y}}$, and bad$\mu$-size-$\mathcal{I}_{ZZ}$. We begin with bounding the size of $|\mathcal{I}_{XX}|$. Let for each $i \in [q]$, $\mathbb{I}_i$ be an indicator random variable that takes the value 1 if there exists a $j \in [q]$ with $j \neq i$ such that $X^i = X^j$. Note that the probability this event holds is at most q/N using the randomness of key $\widehat{R}^i$ (as $i \notin \mathcal{I}_R$), i.e., for a fixed $i \in [q]$,

$$\Pr[\mathbb{I}_i = 1] \leq \frac{q}{N}.$$

Therefore, by the linearity of expectations and by applying Markov's inequality, we have

$$\Pr[|\mathcal{I}_{XX}| > q^{1/2}] \leq \frac{q^{3/2}}{2N}.$$

Similarly, we can show that

$$\Pr[|\mathcal{I}_{\widehat{Y}\widehat{Y}}| > q^{1/2}] \leq \frac{q^{3/2}}{2N}, \quad \Pr[|\mathcal{I}_{ZZ}| > q^{1/2}] \leq \frac{q^{3/2}}{2N}.$$

By combining the above three cases, we have

$$\Pr[\text{bad}\mu\text{-size-coll}] \leq \frac{2q^{3/2}}{N}. \tag{4.144}$$

Finally, by combining Eqn. (4.143) and Eqn. (4.144), we have

$$\Pr[\text{bad}\mu\text{-size}] \leq \frac{q^{1/2}(q_2 + q_3 + q_4)}{N} + \frac{2q^{3/2}}{N}.$$

4.8.12 Bounding badλ-prim

Proposition 12. *Having defined the bad event badλ-prim in Fig. 4.8, we have*

$$\Pr[\textit{bad}\lambda\textit{-prim}] \leq \frac{qq_2(q_1+q_3+q_4+q_5)}{N^2} + \frac{qq_3(q_1+q_2+q_4+q_5)}{N^2} + \frac{qq_4(q_1+q_2+q_3+q_5)}{N^2} + \frac{7q^2(q_2+q_3+q_4)}{N^2}\,.$$

We say that the bad event badλ-prim happens if one of the following events happens.

- badλ-prim 1. $\exists i \in (\mathcal{I}_X \sqcup \mathcal{I}_{**})^c$ and $j \in [q_2]$ such that $\widehat{X}^i + k_2 = V_2^j$.
- badλ-prim 2. $\exists i \in (\mathcal{I}_{\widehat{Y}} \sqcup \mathcal{I}_{**})^c$ and $j \in [q_3]$ such that $Y^i + k_3 = V_3^j$.
- badλ-prim 3. $\exists i \in (\mathcal{I}_Z \sqcup \mathcal{I}_{**})^c$ and $j \in [q_4]$ such that $\widehat{Z}^i + k_4 = V_4^j$.

In the following subsections, we bound the above events.

Bounding badλ-prim 1

To bound this event, we further split it into various sub-cases and bound their probabilities as follows:

- badλ-prim 1*a*. $\exists i \in \mathcal{I}_R$ and $j \in [q_2]$ such that $\widehat{X}^i + K_2 = V_2^j$. In other words, $\exists i \in [q]$, $j \in [q_2]$ and $l \in [q_1]$ such that $R^i + K_1 = U_1^l$ and $\widehat{X}^i + K_2 = V_2^j$. Let's first fix the values for the indices i, j and l. The probability of each of the events $R^i + K_1 = U_1^l$ and $\widehat{X}^i + K_2 = V_2^j$ comes out to be $1/N^2$ each due to the randomness of the keys K_1 and K_2 respectively. As we can choose the index i, j and l in q, q_2 and q_1 ways respectively, we use the union bound over all those possible choices to obtain

$$\Pr[\text{bad}\lambda\text{-prim } 1a] \leq \frac{qq_1q_2}{N^2}\,. \tag{4.145}$$

- badλ-prim 1*b*. $\exists i \in \mathcal{I}_S$ and $j \in [q_2]$ such that $\widehat{X}^i + K_2 = V_2^j$. Analysis of this bad event is similar to that of badλ-prim 1*a*, where we use the randomness of K_5 and K_2. Looking ahead, we bound the probability of the event to at most

$$\Pr[\text{bad}\lambda\text{-prim } 1b] \leq \frac{qq_2q_5}{N^2}\,. \tag{4.146}$$

- badλ-prim 1*c*. $\exists i \in \mathcal{I}_{RR}$ and $j \in [q_2]$ such that $\widehat{X}^i + K_2 = V_2^j$. Analysis of this bad event is similar to that of badλ-prim 1*a*, where we use the randomness of R^i and K_2. Looking ahead, we bound the probability of the event to at most

$$\Pr[\text{bad}\lambda\text{-prim } 1c] \leq \frac{q^2q_2}{2N^2}\,. \tag{4.147}$$

- badλ-prim 1*d*. $\exists i \in \mathcal{I}_{SS}$ and $j \in [q_2]$ such that $\widehat{X}^i + K_2 = V_2^j$. Again, analysis of this bad event is similar to that of badλ-prim 1*c*, where we use the randomness of S^i and K_2. Looking ahead, we bound the probability of the event to at most

$$\Pr[\text{bad}\lambda\text{-prim } 1d] \leq \frac{q^2q_2}{2N^2}\,. \tag{4.148}$$

- badλ-prim 1e. $\exists i \in \mathcal{I}_{\widehat{Y}}$ and $j \in [q_2]$ such that $\widehat{X}^i + K_2 = V_2^j$. In other words, $\exists i \in [q]$, $j \in [q_2]$ and $l \in [q_3]$ such that $\widehat{Y}^i + K_3 = V_3^l$ and $\widehat{X}^i + K_2 = V_2^j$. Let's first fix the values for the indices i, j and l. The probability of each of the events $\widehat{Y}^i + K_3 = V_3^l$ and $\widehat{X}^i + K_2 = V_2^j$ comes out to be $1/N^2$ due to the randomness of the keys K_2 and K_3. As we can choose the index i, j and l in q, q_2 and q_3 ways, we use the union bound over all those possible choices to obtain

$$\Pr[\text{bad}\lambda\text{-prim } 1e] \leq \frac{qq_2q_3}{N^2}\,. \tag{4.149}$$

- badλ-prim 1f. $\exists i \in \mathcal{I}_Z$ and $j \in [q_2]$ such that $\widehat{X}^i + K_2 = V_2^j$. Analysis of this bad event is similar to that of badλ-prim 1e, where we use the randomness of K_4 and K_2. Looking ahead, we bound the probability of the above event to at most

$$\Pr[\text{bad}\lambda\text{-prim } 1f] \leq \frac{qq_2q_4}{N^2}\,. \tag{4.150}$$

- badλ-prim 1g. $\exists i \in \mathcal{I}_{XX}$ and $j \in [q_2]$ such that $\widehat{X}^i + K_2 = V_2^j$. In other words, $\exists i \in [q]$, $j \in [q_2]$ and $l \in [q]$ such that $i \neq l$ and $X^i = X^l, \widehat{X}^i + K_2 = V_2^j$, which we equivalently write as

$$\widehat{R}^i + \widehat{R}^l = L^i + L^l, \widehat{X}^i + K_2 = V_2^j.$$

For a fixed choice of indices, we use the randomness of $\widehat{R}^i$ and K_2 to bound the probability of the event to at most $2/N^2$. As we can choose the index i, j and l in q, q_2 and $(q-1)$ ways, respectively, we use the union bound over all those possible choices to obtain

$$\Pr[\text{bad}\lambda\text{-prim } 1g] \leq \frac{2q^2q_2}{N^2}\,. \tag{4.151}$$

- badλ-prim 1h. $\exists i \in \mathcal{I}_{\widehat{Y}\widehat{Y}}$ and $j \in [q_2]$ such that $\widehat{X}^i + K_2 = V_2^j$. In other words, $\exists i \in [q]$, $j \in [q_2]$ and $l \in [q]$ such that $i \neq l$ and $\widehat{Y}^i = \widehat{Y}^l, \widehat{X}^i + K_2 = V_2^j$, which we equivalently write as

$$\widehat{R}^i + \widehat{R}^l + \widehat{S}^i + \widehat{S}^l = L^i + T^i + L^l + T^l, \widehat{X}^i + K_2 = V_2^j.$$

For a fixed choice of indices, we use the randomness of $\widehat{R}^i$ and K_2 to bound the probability of the event to at most $2/N^2$. As we can choose the index i, j and l in q, q_2 and $(q-1)$ ways respectively, we use the union bound over all those possible choices to obtain

$$\Pr[\text{bad}\lambda\text{-prim } 1h] \leq \frac{2q^2q_2}{N^2}\,. \tag{4.152}$$

- badλ-prim 1i. $\exists i \in \mathcal{I}_{ZZ}$ and $j \in [q_2]$ such that $\widehat{X}^i + k_2 = V_2^j$. In other words, $\exists i \in [q]$, $j \in [q_2]$ and $l \in [q]$ such that $i \neq l$ and $Z^i = Z^l, \widehat{X}^i + K_2 = V_2^j$, which we equivalently write as

$$\widehat{S}^i + \widehat{S}^l = T^i + T^l, \widehat{X}^i + K_2 = V_2^j.$$

For a fixed choice of indices, we use the randomness of $\widehat{S}^i$ and K_2 to bound the probability of the event to at most $2/N^2$. As we can choose the index i, j and l in q, q_2 and $(q-1)$ ways respectively, we use the union bound over all those possible choices to obtain

$$\Pr[\text{bad}\lambda\text{-prim } 1i] \leq \frac{2q^2q_2}{N^2}\,. \tag{4.153}$$

Adding all the above nine cases, we obtain

$$\Pr[\text{bad}\lambda\text{-prim }1] \leq \frac{qq_2(q_1+q_3+q_4+q_5+7q)}{N^2}\,. \tag{4.154}$$

Bounding badλ-prim 2.

As before, to bound this event, we further split it into various sub-cases and bound their probabilities as follows:

- badλ-prim $2a$. $\exists i \in \mathcal{I}_R$ and $j \in [q_3]$ such that $\widehat{Y}^i + K_3 = V_3^j$. In other words, $\exists i \in [q]$, $j \in [q_2]$ and $l \in [q_1]$ such that $R^i + K_1 = U_1^l$ and $\widehat{Y}^i + K_3 = V_3^j$. Let's first fix the values for the indices i, j and l. The probability of each of the events $R^i + K_1 = U_1^l$ and $\widehat{Y}^i + K_3 = V_3^j$ comes out to be $1/N^2$ each due to the randomness of the keys K_1 and K_3 respectively. As we can choose the index i, j and l in q, q_3 and q_1 ways, respectively, we use the union bound over all those possible choices to obtain

$$\Pr[\text{bad}\lambda\text{-prim }2a] \leq \frac{qq_1q_3}{N^2}\,. \tag{4.155}$$

- badλ-prim $2b$. $\exists i \in \mathcal{I}_S$ and $j \in [q_3]$ such that $\widehat{Y}^i + K_3 = V_3^j$. Analysis of this bad event is similar to that of badλ-prim $2a$, where we use the randomness of K_5 and K_3. Looking ahead, we bound the probability of the event to at most

$$\Pr[\text{bad}\lambda\text{-prim }2b] \leq \frac{qq_3q_5}{N^2}\,. \tag{4.156}$$

- badλ-prim $2c$. $\exists i \in \mathcal{I}_{RR}$ and $j \in [q_3]$ such that $\widehat{Y}^i + K_3 = V_3^j$. Analysis of this bad event is similar to that of badλ-prim $2a$, where we use the randomness of R^i and K_3. Looking ahead, we bound the probability of the event to at most

$$\Pr[\text{bad}\lambda\text{-prim }2c] \leq \frac{q^2q_3}{2N^2}\,. \tag{4.157}$$

- badλ-prim $2d$. $\exists i \in \mathcal{I}_{SS}$ and $j \in [q_3]$ such that $\widehat{Y}^i + K_3 = V_3^j$. Analysis of this bad event is similar to that of badλ-prim $2c$, where we use the randomness of S^i and K_3. Looking ahead, we bound the probability of the event to at most

$$\Pr[\text{bad}\lambda\text{-prim }2d] \leq \frac{q^2q_3}{2N^2}\,. \tag{4.158}$$

- badλ-prim $2e$. $\exists i \in \mathcal{I}_Z$ and $j \in [q_3]$ such that $\widehat{Y}^i + K_3 = V_3^j$. Analysis of this bad event is again similar to that of badλ-prim $1f$, where we use the randomness of K_4 and K_3. Looking ahead, we bound the probability of the event to at most

$$\Pr[\text{bad}\lambda\text{-prim }2e] \leq \frac{qq_3q_4}{N^2}\,. \tag{4.159}$$

- badλ-prim $2f$. $\exists i \in \mathcal{I}_X$ and $j \in [q_3]$ such that $\widehat{Y}^i + K_3 = V_3^j$. Analysis of this bad event is again similar to that of badλ-prim $2a$, where we use the randomness of K_2 and K_3. Looking ahead, we bound the probability of the event to at most

$$\Pr[\text{bad}\lambda\text{-prim }2f] \leq \frac{qq_2q_3}{N^2}\,. \tag{4.160}$$

- badλ-prim $2g$. $\exists i \in \mathcal{I}_{XX}$ and $j \in [q_3]$ such that $\widehat{Y}^i + K_3 = V_3^j$. Analysis of this event is similar to that of badλ-prim $1g$, where we use the randomness of $\widehat{R}^i$ and K_3. Looking ahead, we bound the probability of the event to at most

$$\Pr[\text{bad}\lambda\text{-prim } 2g] \leq \frac{2q^2 q_3}{N^2}\,. \tag{4.161}$$

- badλ-prim $2h$. $\exists i \in \mathcal{I}_{\widehat{Y}\widehat{Y}}$ and $j \in [q_3]$ such that $\widehat{Y}^i + K_3 = V_3^j$. Analysis of this event is similar to that of badλ-prim $1h$, where we use the randomness of $\widehat{R}^i$ and K_3. Looking ahead, we bound the probability of the event to at most

$$\Pr[\text{bad}\lambda\text{-prim } 2h] \leq \frac{2q^2 q_3}{N^2}\,. \tag{4.162}$$

- badλ-prim $2i$. $\exists i \in \mathcal{I}_{ZZ}$ and $j \in [q_3]$ such that $\widehat{Y}^i + K_3 = V_3^j$. Again, the analysis of this event is similar to that of badλ-prim $1i$, where we use the randomness of $\widehat{S}^i$ and K_3. Looking ahead, we bound the probability of the event to at most

$$\Pr[\text{bad}\lambda\text{-prim } 2i] \leq \frac{2q^2 q_3}{N^2}\,. \tag{4.163}$$

Adding all the above nine cases, we obtain

$$\Pr[\text{bad}\lambda\text{-prim } 2] \leq \frac{q q_3 (q_1 + q_2 + q_4 + q_5 + 7q)}{N^2}\,. \tag{4.164}$$

Bounding badλ-prim 3.

As before, to bound this event, we further split it into various sub-cases and bound their probabilities as follows:

- badλ-prim $3a$. $\exists i \in \mathcal{I}_R$ and $j \in [q_4]$ such that $\widehat{Z}^i + K_4 = V_4^j$. In other words, $\exists i \in [q]$, $j \in [q_4]$ and $l \in [q_1]$ such that $R^i + K_1 = U_1^l$ and $\widehat{Z}^i + K_4 = V_4^j$. Let's first fix the values for the indices i, j and l. The probability of each of the events $R^i + K_1 = U_1^l$ and $\widehat{Z}^i + K_4 = V_4^j$ comes out to be $1/N^2$ each due to the randomness of the keys K_1 and K_4 respectively. As we can choose the index i, j and l in q, q_4 and q_1 ways respectively, we use the union bound over all those possible choices to obtain

$$\Pr[\text{bad}\lambda\text{-prim } 3a] \leq \frac{q q_1 q_4}{N^2}\,. \tag{4.165}$$

- badλ-prim $3b$. $\exists i \in \mathcal{I}_S$ and $j \in [q_4]$ such that $\widehat{Z}^i + K_4 = V_4^j$. Analysis of this bad event is similar to that of badλ-prim $3a$, where we use the randomness of K_5 and K_4. Looking ahead, we bound the probability of the event to at most

$$\Pr[\text{bad}\lambda\text{-prim } 3b] \leq \frac{q q_4 q_5}{N^2}\,. \tag{4.166}$$

- badλ-prim $3c$. $\exists i \in \mathcal{I}_{RR}$ and $j \in [q_4]$ such that $\widehat{Z}^i + K_4 = V_4^j$. Analysis of this bad event is similar to that of badλ-prim $3a$, where we use the randomness of R^i and K_4. Looking ahead, we bound the probability of the event to at most

$$\Pr[\text{bad}\lambda\text{-prim } 3c] \leq \frac{q^2 q_4}{2N^2}\,. \tag{4.167}$$

- badλ-prim $3d$. $\exists i \in \mathcal{I}_{SS}$ and $j \in [q_4]$ such that $\widehat{Z}^i + K_4 = V_4^j$. Analysis of this bad event is similar to that of badλ-prim $3a$, where we use the randomness of S^i and K_4. Looking ahead, we bound the probability of the event to at most
$$\Pr[\text{bad}\lambda\text{-prim } 3d] \leq \frac{q^2 q_4}{2N^2} \,. \tag{4.168}$$

- badλ-prim $3e$. $\exists i \in \mathcal{I}_{X}$ and $j \in [q_4]$ such that $\widehat{Z}^i + K_4 = V_4^j$. Analysis of this bad event is similar to that of badλ-prim $3a$, where we use the randomness of K_2 and K_4. Looking ahead, we bound the probability of the event to at most
$$\Pr[\text{bad}\lambda\text{-prim } 3e] \leq \frac{q q_2 q_4}{N^2} \,. \tag{4.169}$$

- badλ-prim $3f$. $\exists i \in \mathcal{I}_{\widehat{Y}}$ and $j \in [q_4]$ such that $\widehat{Z}^i + K_4 = V_4^j$. Analysis of this bad event is similar to that of badλ-prim $3a$, where we use the randomness of K_3 and K_4. Looking ahead, we bound the probability of the event to at most
$$\Pr[\text{bad}\lambda\text{-prim } 3f] \leq \frac{q q_3 q_4}{N^2} \,. \tag{4.170}$$

- badλ-prim $3g$. $\exists i \in \mathcal{I}_{XX}$ and $j \in [q_4]$ such that $\widehat{Z}^i + K_4 = V_4^j$. Analysis of this bad event is similar to that of badλ-prim $1g$, where we use the randomness of $\widehat{R}^i$ and K_4. Looking ahead, we bound the probability of the event to at most
$$\Pr[\text{bad}\lambda\text{-prim } 3g] \leq \frac{2q^2 q_4}{N^2} \,. \tag{4.171}$$

- badλ-prim $3h$. $\exists i \in \mathcal{I}_{\widehat{Y}\widehat{Y}}$ and $j \in [q_4]$ such that $\widehat{Z}^i + K_4 = V_4^j$. Analysis of this bad event is similar to that of badλ-prim $1h$, where we use the randomness of $\widehat{R}^i$ and K_4. Looking ahead, we bound the probability of the event to at most
$$\Pr[\text{bad}\lambda\text{-prim } 3h] \leq \frac{2q^2 q_4}{N^2} \,. \tag{4.172}$$

- badλ-prim $3i$. $\exists i \in \mathcal{I}_{ZZ}$ and $j \in [q_4]$ such that $\widehat{Z}^i + K_4 = V_4^j$. Analysis of this bad event is similar to that of badλ-prim $1i$, where we use the randomness of $\widehat{S}^i$ and K_4. Looking ahead, we bound the probability of the event to at most
$$\Pr[\text{bad}\lambda\text{-prim } 3i] \leq \frac{2q^2 q_4}{N^2} \,. \tag{4.173}$$

Adding all the above nine cases, we obtain
$$\Pr[\text{bad}\lambda\text{-prim } 3] \leq \frac{q q_4 (q_1 + q_2 + q_3 + q_5 + 7q)}{N^2} \,. \tag{4.174}$$

4.8.13 Bounding badλ-coll

Proposition 13. *Having defined the bad event badλ-coll in Fig. 4.8, we have*
$$\Pr[\textit{bad}\lambda\textit{-coll}] \leq \frac{\binom{q}{2}(5q + q_1 + q_2 + q_3 + q_4 + q_5)}{N^2} \,.$$

We say that the bad event badλ-coll happens if one of the following events happens.

- badλ-coll 1. $\exists i \in \mathcal{I}^c_{**}, j \in [q]$ and $i \neq j$ such that $X^i \neq X^j$ and $\widehat{X}^i = \widehat{X}^j$.
- badλ-coll 2. $\exists i \in \mathcal{I}^c_{**}, j \in [q]$ and $i \neq j$ such that $\widehat{Y}^i \neq \widehat{Y}^j$ and $Y^i = Y^j$.
- badλ-coll 3. $\exists i \in \mathcal{I}^c_{**}, j \in [q]$ and $i \neq j$ such that $Z^i \neq Z^j$ and $\widehat{Z}^i = \widehat{Z}^j$.

In the following subsection, we bound the above events. To do this, we first define a condition set and then analyse these three bad events on that condition set.

Condition Set

1. $\exists i \in \mathcal{I}_R$. In other words, $\exists i \in [q]$ and $k \in [q_1]$ such that $R^i + K_1 = U_1^k$.
2. $\exists i \in \mathcal{I}_S$. In other words, $\exists i \in [q]$ and $k \in [q_5]$ such that $S^i + K_5 = U_5^k$.
3. $\exists i \in \mathcal{I}_{RR}$. In other words, $\exists i \in \mathcal{I}_{\mathsf{dec}}$ and $k \in [i-1]$ such that $R^i = R^k$.
4. $\exists i \in \mathcal{I}_{SS}$. In other words, $\exists i \in \mathcal{I}_{\mathsf{enc}}$ and $k \in [i-1]$ such that $S^i = S^k$.
5. $\exists i \in \mathcal{I}_X$. In other words, $\exists i \in [q]$ and $k \in [q_2]$ such that $X^i + K_2 = U_2^k$.
6. $\exists i \in \mathcal{I}_{\widehat{Y}}$. In other words, $\exists i \in [q]$ and $k \in [q_3]$ such that $\widehat{Y}^i + K_3 = U_3^k$.
7. $\exists i \in \mathcal{I}_Z$. In other words, $\exists i \in [q]$ and $k \in [q_4]$ such that $Z^i + K_4 = U_4^k$.
8. $\exists i \in \mathcal{I}_{XX}$. In other words, $\exists i, k \in [q]$ with $i \neq j$ such that $X^i = X^k$.
9. $\exists i \in \mathcal{I}_{\widehat{Y}\widehat{Y}}$. In other words, $\exists i, k \in [q]$ with $i \neq j$ such that $\widehat{Y}^i = \widehat{Y}^k$.
10. $\exists i \in \mathcal{I}_{ZZ}$. In other words, $\exists i, k \in [q]$ with $i \neq j$ such that $Z^i = Z^k$.

Let's first fix the values for the indices i, j and k. For any of badλ-coll 1, badλ-coll 2 and badλ-coll 3, any one of the conditions from the above condition set satisfies. Once we fix that condition, the probability of that condition comes out to be $(1/N)$. On the other hand, the probability of the event $\widehat{X}^i = \widehat{X}^j$ is at most $(2/N)$ when $j \in \mathcal{I}_X$, and is equal to $(1/N)$ otherwise. Similarly, the probability of the event $Y^i = Y^j$ is at most $(2/N)$ when $j \in \mathcal{I}_Y$, and is equal to $(1/N)$ otherwise; and the probability of the event $\widehat{Z}^i = \widehat{Z}^j$ is at most $(2/N)$ when $j \in \mathcal{I}_Z$, and is equal to $(1/N)$ otherwise. Now one can choose the pair of indices (i, j) in $\binom{q}{2}$ ways and the index k in as many ways as the maximum number of queries to the relevant permutation (in case of conditions 1, 2, 5, 6 and 7) or in q ways (otherwise). Using the union bound over all those possible indices, we obtain the upper bound of each of these bad events as $(2q \cdot \binom{q}{2})/(N^2)$ or $(2q_l \cdot \binom{q}{2})/(N^2)$ (where the relevant permutation is P_l).

Chapter 5

Crooked Indifferentiability of Enveloped Xor

5.1 Introduction

In CRYPTO 2018, Russell, Tang, Yung, and Zhou (RTYZ) introduced the notion of crooked indifferentiability to analyse the security of a hash function when the underlying primitive is subverted. They showed that the n-bit to n-bit function implemented using enveloped XOR construction (EXor) with $3n+1$ many n-bit functions and $3n^2$-bit random initial vectors can be proven secure asymptotically in the crooked indifferentiability setting. We identify several major issues and gaps in the proof by RTYZ; we argue that their proof can achieve security only in a restricted setting. We present a new proof of crooked indifferentiability where the adversary can evaluate queries related to multiple messages. Our technique can handle function-dependent subversion.

Random Oracle and Indifferentiability. The *Random Oracle* methodology is a very popular platform for proving the security of cryptographic constructions in the black-box fashion. In this model, all the parties, including the adversary, are given access to a common truly random function. One proves the security of a protocol, assuming such a random function exists. During the implementation of the protocol, the random function is replaced by a hash function H. The *Indifferentiability* framework and the composition theorem [67] assert that if the hash function H is based on an ideal primitive f, and Indifferentiable from a random function, then the instantiated protocol is as secure as the protocol in the random oracle model (assuming the security of the ideal primitive f). Indifferentiability from Random Oracle has been one of the mainstream security criteria of cryptographic hash functions. Starting from the work of Coron, Dodis, Malinaud, and Puniya [36], a plethora of results [28, 18, 19, 69, 73] have been proven, showing the indifferentiability of different constructions based on different ideal primitives.

Crooked Indifferentiability. In CRYPTO 2018, Russel, Tang, Yung and Zhou [90] introduced the notion of crooked indifferentiability as a security notion for hash functions in the kleptographic setting. Like classical indifferentiability, the game of crooked indifferentiability challenges the adversary to distinguish between two worlds. In the real world, the adversary has access to the underlying ideal primitive f, and the construction C, which has subroutine access to $\tilde{f}$, the subverted implementation of f.[1] The implementation $\tilde{f}$ on input an element x, queries the function (possibly adaptively) at maximum $\tilde{q}$ many points and, based on the transcript,

[1] The domain extension algorithms are simple and the correctness of their implementations is easy to verify.

decides the evaluation of x. As the adversary likes the subversion to go undetected, it is assumed that $\tilde{f}$ differs from f only on some negligible fraction (ϵ) of the domain.

In the ideal world, the construction is replaced by a Random Oracle $\mathcal{F}$. The role of f is played by a simulator with Oracle access to $\mathcal{F}$ and the subverted implementation $\tilde{f}$. The job of the simulator is to simulate f in such a way that $(C^{\tilde{f}}, f)$ is indistinguishable from $(\mathcal{F}, S^{\mathcal{F},\tilde{f}})$. To avoid trivial attacks, the framework allows a *public* random string R to be used as the salt in the construction. The string R is fixed after the adversary publishes the implementation but stays the same throughout the interaction. All the parties, including the simulator and the adversary, get R as part of the initialisation input. We note that even in the weaker setting of Random Oracles with auxiliary input, a random salt is required to prove security [35, 49].

ENVELOPED XOR CONSTRUCTION AND ITS CROOKED INDIFFERENTIABILITY. We recall the Enveloped XOR construction. We fix two positive integers n and l. Let $\mathcal{D} := \{0, 1, \ldots, l\} \times \{0,1\}^n$. Let H be the class of all functions $f : \mathcal{D} \to \{0,1\}^n$. For every $x \in \{0,1\}^n$ and an initial value $R := (r_1, \ldots, r_l) \in (\{0,1\}^n)^l$, we define

$$g_R(x) = \bigoplus_{i=1}^{l} f(i, x \oplus r_i) \quad \text{and} \quad \mathsf{EXor}(R, x) = f(0, g_R(x)).$$

In [90], Russell et al. proved crooked indifferentiability of the enveloped-xor construction. Their analysis is based on an interesting rejection sampling argument.

5.2 Our Contribution

Another Look at Russell et al.'s Proof. We uncover that the techniques of [90], while novel and interesting, bear significant shortcomings. The consistency of the simulator is not proven. Moreover, their technical treatment requires that the subversion for the final function $f(0, \cdot)$ be independent of g_R. In other words, the proof is applicable against a restricted class of subversion. Finally, the proof does not consider the messages queried to $\mathcal{F}$. We elaborate on the issues in Section 5.4.

A New Proof of the Crooked Indifferentiability of Enveloped XOR. We present a new proof of the crooked indifferentiability of Enveloped XOR. Interestingly, our techniques do not involve heavy technical machinery. Rather, we identify core domain points related to functions and use simple tools like Markov inequality.

5.2.1 Overview of Our Technique.

We observe the Enveloped XOR (EXoR) construction is in the class of Generalised Domain Extensions considered in [18]. It is known that for a GDE construction with independent functions and preimage awareness, the indifferentiability advantage is bounded by the probability that the final chaining query is not fresh. However, EXoR construction instantiated with the crooked functions (denoted by $\widetilde{\mathsf{EXor}}$) is not part of GDE. The main issue is that the final output of $\widetilde{\mathsf{EXor}}$ need not be the output of $f(0, \cdot)$ evaluation, as required by the condition of GDE.

We consider an intermediate construction $\overline{\mathsf{EXor}}(R, m) = f(0, \tilde{g}_R(m))$. In other words, the intermediate construction restricts that the finalization

function $f(0, \cdot)$ is not subverted. $\overline{\mathsf{EXor}}$ is a GDE construction and crooked indifferentiability of $\overline{\mathsf{EXor}}$ can be proved following the structure of [18]. In particular, the generic simulator of [18], adopted for $\overline{\mathsf{EXor}}$ along with access to $\tilde{f}$ work out here along with the consistency arguments. Our proof is modularised via the following two claims.

- Claim 2 shows distinguishing advantage for $(f, \overline{\mathsf{EXoR}})$ and $(f, \widetilde{\mathsf{EXoR}})$ is bounded by the probability of hitting a crooked point or domain point for f_0 (Bad1).

- Claim 3 shows the distinguishing advantage of the intermediate world $(f, \overline{\mathsf{EXoR}})$, and the ideal world of crooked indifferentiability is bounded by the probability of Bad2 event. This event is classified into two main categories. In the first category, while responding to a query to the primitive (or the simulator), input $\tilde{g}_R(m)$ appeared already in the transcript. In the second category, the input $\tilde{g}_R(m)$ appeared in the extended transcript which includes all queries of a subverted computation $\tilde{f}(x)$ of a crooked point x.

The challenge remaining is to bound the probability of the bad events. Our proof works with a counting approach. We say a point $\alpha \in \{0,1\}^n$ is robust with respect to a function f if all points that query α are not subverted with all but negligible probability if the output $f(\alpha)$ is re-sampled. A point is good if it is queried by only a few robust and un-crooked points. By an averaging argument, we show that for an overwhelming fraction of candidate f, R, for every message m, there will exist an index i such that mr_i is good for function $f(i,\cdot)$. Now, we can say that even though $f(i, mr_i)$ was queried by other points, they are robust. If we re-sample at (i, mr_i), the subverted outputs of those robust points will not change. Thus, we can talk about $\tilde{g}_R(m)$ independently of the outputs of the function $\tilde{f}(0,\cdot)$.

Finally, we shall show that the output distribution of $\tilde{g}_R(m)$ is close to uniform. We could find a rejection resampling lemma on two or more points, and argue the uniformity of $\tilde{g}_R(m)$. However, we simplify things further. We observe that with high probability over the output values of $f(i, mr_i)$ for every i for which mr_i is good in f, the transcript of the previous internal queries remains unchanged. Hence, we consider the conditional probabilities by conditioning on all possible transcripts and take union bound to show near uniformity of $\tilde{g}_R(m)$.

Relation of GDE Constructions with Our Results and Further Uses.

A majority of this work focuses on $\overline{\mathsf{EXor}}$ construction, which is a GDE construction (defined in [18]). GDE constructions cover a wide range of domain extension algorithms. We believe that many ideas developed in this result to deal with the $\overline{\mathsf{EXor}}$ construction can be extended to investigate the crooked indifferentiability of different GDE constructions. However, the bad events and their analysis will depend on the particular construction being investigated.

Revised Proof by Russell et al.

After we communicated our findings to the authors of [90], they acknowledged the issues, and uploaded a major revision in eprint [91]. Our proof is done independently and significantly differs from their revised proof in some crucial aspects.

5.3 Recalling the Proof of Russell et al.

5.3.1 Enveloped XOR Construction.

Recall that, in the real world, the distinguisher is interacting with the subverted construction $\widetilde{\mathsf{EXor}}$ which is defined as

$$\widetilde{\mathsf{EXor}}(R, M) = \tilde{f}(0, \tilde{g}_R(M)) \quad \text{where} \quad \tilde{g}_R(M) = \bigoplus_{i=1}^{l} \tilde{f}(i, M \oplus r_i).$$

We also define a hybrid construction $\overline{\mathsf{EXor}}[f](R, M) = f(0, \tilde{g}_R(M))$. Now consider an adversary $\mathcal{A}$ interacting with $(f, \overline{\mathsf{EXor}} := \overline{\mathsf{EXor}}[f])$.

Assumption on Adversary. For all primitive queries of the form (j, x) with $j > 0$, we return $\overline{\mathsf{EXor}}(m)$ and all responses of all queries $(a, \alpha_a), a \in \underline{[l]}$ where $\alpha_a = m + R_a$ and $m = x + R_j$. Note that the simulator can compute m, so responding $\overline{\mathsf{EXor}}(m)$ honestly for the simulator would not be a problem. Moreover, we assume that the adversary discloses all queries for the construction to the simulator.

Transcript of Interaction. For $j \geq 0$, let $\tau_j := (R, \tau_j, \pi_j)$ denote the transcript (random variable due to randomness of f only) of $\mathcal{A}$ after j queries where R is the initial value of the construction, and τ_j, π_j denote the query-responses for the primitive and the construction respectively. Note that τ_j contains τ_0 for all j.

5.4 Revisiting the Crooked Indifferenitability Security of EXoR [90].

A Brief Detour: Classical Indifferentiability Simulator for EXor.

Before describing the crooked indifferentiability simulator, we would like to briefly recall the principle behind the indifferentiability simulator and proof principles behind EXor construction in the classical setting.

The goal of the simulator is to simulate each $f(i, \cdot)$ honestly so that for every queried message m, it holds that $\mathsf{EXor}(R, m) = \mathcal{F}(m)$ for all queried m. Without loss of generality, assume that whenever the adversary makes queries $f(i, x)$ for $i > 0$, it also makes queries $f(j, x \oplus r_i \oplus r_j)$ for all $j > 0$ simultaneously. In other words, it makes a batch query of the form $(f(j, m \oplus r_j))_{1 \leq j \leq l}$ for some $m \in \{0, 1\}^n$. We simply say that the adversary $\mathcal{A}$ queries m to g_R and obtains responses $(f(j, m \oplus r_j))_{1 \leq j \leq l}$. On receiving a batch query $g_R(m)$, the simulator will honestly sample outputs for the corresponding $f(i, mR_i)$ queries for all $i \in (l]$, and compute $g_R(m)$ by xoring those sampled outputs. Also, the simulator will save the queried m along with the computed $g_R(m)$ in a list L. For a $f(0, x)$ query, the simulator will first search in L, whether for some m, it has given $x = g_R(m)$ as output. If yes, the simulator simply returns $\mathcal{F}(m)$. If no such entry exists, the simulator samples an output z uniformly at random and returns z.

Now, we briefly recall how the indifferentiability is proved for this simulator. There are two bad events.

- for distinct m, m', it holds that $g_R(m) = g_R(m')$. In this case, the simulator, on query $f(0, g_R(m))$ can not be consistent with both $\mathcal{F}(m)$ and $\mathcal{F}(m')$ with any significant probability.

- For a batch query $g_R(m)$ the output is such that it matches with a previous $f(0, .)$ query. In this case, the simulator has already given output to the $f(0, .)$ query, which, with all but negligible probability, is not equal to $\mathcal{F}(m)$.

One can indeed summarise these bad events as one; $g_R(m) \in E$, where E is the set of $f(0, .)$ queries made by the adversary.

The Simulator for Crooked Indifferentiability.

We now describe the main idea behind the simulator in the crooked indifferentiability setting. The same principle was used in [90]. Note, here, the main goal of the simulator is different. It needs to simulate $f \leftarrow\!\!\$\ \mathsf{H}$ as honestly [2] as possible such that $\widetilde{\mathsf{EXor}}(R, m) = \mathcal{F}(m)$ for all queried m. Thus, the simulator needs to ensure that the output of the random oracle matches with the *subverted implementation* of EXor.

The simulator maintains a list L of pairs (α, β) to record $f(\alpha) = \beta$ for $\alpha \in \mathcal{D}$ and $\beta \in \{0,1\}^n$. It also maintains a sub-list $L^A \subseteq L$ consisting of all those pairs which are known to the distinguisher. Both lists are initialised to z (the advice string in the first stage which we fix to any tuple of q_1 pairs). $L_0 = L_0^A = z$. Now we describe how the simulator responds.

1. (Query $f(0,w)$) We call this query a Type-1 Query. Type-1 Queries are returned honestly. If $((0,w), y) \in L$ for some y, the simulator returns the same y. Otherwise, it samples y uniformly from $\{0,1\}^n$, updates the list L and L^A, and returns y.

2. (Query $g_R(m)$) We call this Type-2 Query. For a query $g_R(m)$ (i.e. batch query) the simulator computes $\tilde{f}(\alpha_j)$ for all j, one by one by executing the subverted implementation $\tilde{F}$, where $\alpha_j = (j, m \oplus R_j)$. During this execution, the simulator responds honestly to all queries made by the subverted implementation and updates the L-list by incorporating all query responses of h. However, it updates L^A list only with $(\alpha_j, f(\alpha_j))$ for all j. Let $\tilde{\mathsf{g}} := \bigoplus_j \tilde{f}(\alpha_j)$. If $(0, \tilde{\mathsf{g}}) \in \mathcal{D}(L)$, the simulator **aborts** . If the simulator does not abort, it makes a query $\mathcal{F}(m)$ and adds $((0, \tilde{\mathsf{g}}), \mathcal{F}(m))$ into both lists L and L^A.

For $f(0,w)$ made by $\mathcal{A}_2$ where $w = \tilde{g}_R(m)$ for some previous query m to g_R, the simulator responds as $\mathcal{F}(m)$.

Cautionary Note. Even though $\mathcal{F}$ is a random oracle, we cannot say that the probability distribution of the response of $(0, \tilde{\mathsf{g}})$ in the ideal world is uniform. Note that, the adversary can choose m after making several consultations with $\mathcal{F}$. In other words, m can be dependent on $\mathcal{F}$. For example, the adversary can choose a message m for which the last bit of $\mathcal{F}(m)$ is zero. Thus, the response for the query $(0, w)$ always has zero as the last bit (which diverts from the uniform distribution). However, the randomness can be considered when we consider the joint probability distribution of all query responses.

Transcript: Now we describe what is the transcript to the distinguisher and for the simulator in more detail. First, we introduce some more relevant notations.

1. Let L^F denote the set of all pairs $(m', \tilde{z})$ of query response of $\mathcal{F}$ by $\mathcal{A}_2$.

2. Let L^g denote the set of all pairs (m, β^l) of query response of g_R oracle (batch query) made by $\mathcal{A}_2$ to the simulator where $\beta^l := (\beta_1, \ldots, \beta_l)$ and $\beta_j = h(j, m \oplus R_j)$ for all j. According to our convention, all these m must be queried to $\mathcal{F}$ beforehand.

3. As we described, we also have two lists, namely L and its sublist L^A, keeping the query responses of h oracle.

Now, we define the transcript and partial transcript of the interaction. We recall that q_1 is the number of queries in the first stage and $\mathcal{A}_2$ is a (q_F, q_2)-query algorithm. Let $q = q_2 + q_F$ For any $1 \le i \le q$, we define the partial transcript of $\mathcal{A}$ and the simulator as $\tau_i^A := (L_i^F, L_i^A)$ and $\tau_i^S := (L_i, L_i^g)$ respectively, where L_i^F, L_i^A, L_i, L_i^g denote the contents of the corresponding lists just before making i-th query of the distinguisher. So when, $i = 1$, $L_1^A = L_1 = z$ and the rest are empty and when $i = q + 1$, these are the final lists of transcripts. Let $\tau_i := (\tau_i^A, \tau_i^S)$ and

[2] By honestly we mean perfectly simulating a random function. If the responses are already in the list it returns that value; otherwise, it samples a fresh random response and includes the input and output pairs in the list.

$\tau := (\tau^A, \tau^S)$ denote the joint transcript on i-th query or after completion respectively. As the adversary is deterministic, the simulator is also deterministic for a given h and $\mathcal{F}$, and we have fixed z, a (partial) transcript is completely determined by the choice of R, h and $\mathcal{F}$ (in the ideal world). We write $(R, f, \mathcal{F}) \vdash \tau_i^S$ if the transcript τ_i^S is obtained when the initial value is R, the random oracles are $\mathcal{F}$ and f. We similarly define $(R, f, \mathcal{F}) \vdash \tau_i^A$ and $(R, f, \mathcal{F}) \vdash \tau_i$.

5.4.1 Techniques of [90]

Overview of the Techniques in [90]. We assume, without any loss of generality, that the second stage adversary $\mathcal{A}_2$ queries m to $\mathcal{F}$ before it queries to g_R oracle. In addition, like before, we assume that it makes batch queries.

For every query number i, we define a set $E_i := \mathcal{D}(L_i) \cup \mathsf{subv}_f$ where subv_f is the set of all crooked elements for f. The event bad_i holds if and only if $(0, \tilde{g}_R(m_i)) \in E_i$ where m_i denotes the i-th query of $\mathcal{A}$ (made to g_R oracle of the simulator). So, the crooked indifferentiable advantage is bounded by $\sum_{i=1}^{q_2} \Pr(\tilde{g}_R(m_i) \in E_i)$. The authors wanted to show that the distribution of $\tilde{g}_R(m_i)$ is almost uniform. They proposed the following theorem.

(**Theorem 5** from [90]). With overwhelming probability (i.e., one minus a negligible amount) there exists a set $\mathcal{R}_{\tau_0} \subseteq (\{0,1\}^n)^l$ and for every i, a set of transcripts $\mathcal{T}_i^A$ (before i-th query) such that for all $R \in \mathcal{R}_{\tau_0}$, $\tau_i := (L_i^F, L_i^A) \in \mathcal{T}_i^A$, and $m \notin \mathcal{D}(L_i^g)$,

$$\Pr_{\mathsf{f}}[(0, \tilde{g}_{\mathsf{R}}(m)) \in E_i \mid (R, \mathsf{f}, \mathcal{F}) \vdash \tau_i] \le \mathrm{poly}(n)\sqrt{|E_i|} + \mathsf{negl}(n).$$

The authors claimed that the crooked indifferentiability of EXor can be derived from the above theorem. To describe the issues we need to dive into the main idea which is to show that $\tilde{g}_{\mathsf{R}}(m)$ behaves close to the uniform distribution over $\{0,1\}^n$. Thus the above probability would be negligible as $q_1/2^n$ and $|\mathsf{subv}_f|/2^n$ is negligible. By using Markov inequality, authors are able to identify a set of overwhelming amount of pairs (R, f), called *unpredictable* pair, such that for any unpredictable (R, f) all m, there exists an index i such that

1. $\Pr_\beta[\alpha_i \in \mathsf{subv}_f \mid f(\alpha_i) = \beta]$ is negligible and
2. $\alpha_j \notin Q_f^{-1}(\alpha_i)$ for all $j \neq i$, where $\alpha_j = m \oplus R_i$.

Thus, if we resample $\beta = f(\alpha_i)$ then with overwhelming probability $\tilde{f}|_{\alpha_i \to \beta}(\alpha_i) = f|_{\alpha_i \to \beta}(\alpha_{)}$ (i.e. α_i is not crooked and returned a random value) and all corresponding values for indices j different from i will remain the same. So, $\tilde{g}_R(m) = \beta + A$ where A does not depend on choice of β. Thus, the modified distribution is close to uniform (as almost all values of β will be good). In particular, the authors made the following claim:

Claim 1. *Under the modified distribution (i.e. after resampling),* $\Pr[\tilde{g}_R(m) \in E_1] \le q_1/2^n + \epsilon + p_n$ *where* p_n *denotes the probability that a random pair* (R, f) *is not unpredictable.*

As the choice of i depends on the function f and so a new rejection resampling lemma is used to bound the probability of the event under the original distribution (i.e. before resampling).

Lemma 10 (Rejection Resampling [90])**.** *Let* $X := (X_1, \ldots, X_k)$ *be a random variable uniform on* $\Omega = \Omega_1 \times \Omega_2 \times \cdots \times \Omega_k$*. Let* $A : \Omega \to (k]$ *and define* $Z = (Z_1, \ldots, Z_k)$ *where* $Z_i = A_i$ *except at* $j = A(X^k)$ *for which* Z_j *is sampled uniformly and independently of remaining random variables. Then for any event* $S \subseteq \Omega$*, it holds that*

$$|S|/|\Omega| \le \sqrt{k \Pr(Z \in S)}$$

With this rejection resampling result and Claim 1, the authors concluded the following under original distribution:

$$\Pr_{h*}(\tilde{g}_R(x) \in E_1) \leq \sqrt{l \cdot \Pr_{\text{resampled } h}(\tilde{g}_R(x) \in E_1)} \leq \sqrt{l \cdot (q_1/2^n + \epsilon + p_n)}.$$

5.4.2 Issues with the Technique of [90]

Now we are ready to describe the issues and the limitations of the techniques in [90]. To prove the general case (i.e. for any query), the authors provide a proof sketch where they argue that with an overwhelming probability of realizable transcript $\mathcal{T}$ and for all $\tau \in \mathcal{T}$, $\Pr(\tilde{g}_R(m_i) \in E_i \mid \tau)$ is negligible.

The Number of Queries to $\mathcal{F}$ is Essential. An incompleteness of the proof of [90] comes from the fact that the analysis does not consider the $\mathcal{F}$ queries of the distinguisher. The bound is almost vanishing if $q_1 = 0$ and $q_2 = 2$ and there is no crooked point. However, a distinguisher can search for $m \neq m'$ such that $\mathcal{F}(m) = \mathcal{F}(m')$. Conditioned on collision at the final output, the event $g_R(m) = g_R(m')$ holds with probability of about $1/2$. On the other hand, for the honest simulation of all f values, the g value will collide with a very low probability. If the adversary can make $2^{n/2}$ many queries to $\mathcal{F}$, the above inconsistency can be forced. Hence, *the probability upper bound of Theorem 5 of [90] can not be independent of the number of queries made to $\mathcal{F}$.*

Inconsistency for Multiple Queries: Controlling Query Dependencies for the Same Index. Authors claimed that for all unpredictable (R, h), for all m, an index i exists on which the resampling can be done *without affecting the transcript.* Recalling the notion of unpredictable (R, h) we see that the resampling is done on an index i, that is honest ($\tilde{f}(i, mR_i) = f(i, mR_i)$), and $f(i, mR_i)$ is not queried by $f(j, mR_j)$ for any other j. From here, the authors argued that the transcript of the interaction remains the same if we resample at such i. This claim is justified for a single message and not for multiple queries. We note that it is easy to construct a subverted implementation $\tilde{F}$ for which all inputs of f for a batch response are queried during some other previous query. For example, if it queries $f(i, x1^n)$ for an input (i, x), and the distinguisher makes two batch queries, $\tilde{g}_R(m1^n)$, and $\tilde{g}_R(m)$. The simulator, while simulating $\tilde{g}_R(m1^n)$ responds to all the queries made by $\tilde{f}(i, m1^n R_i)$, and in particular the value of $f(i, mR_i)$ is now gets fixed. *So, an appropriate analysis was missing in case of multiple queries.*

The Bad Event E_i Depends on the Function f. The main technical claim of [90] that $\Pr_{\text{resampled } f}(\tilde{g}_R(x) \in E)$ is small because $\tilde{g}_R(x)$ is uniformly distributed under the resampling distribution of f and the size of E is negligibly small. However, the crooked set of $f(0, \cdot)$ may depend on the other functions $f(1, \cdot), \ldots, f(\cdot)$. Thus the event E is not independent of $\tilde{g}_R(x)$. In particular, one cannot upper bound the $\Pr(\tilde{g}_R(x) \in E)$ as $|E|/2^n$. This is one of the crucial observations which makes the crooked security analysis quite a complex task.

5.5 Basic Setup: Good Pairs and Critical Set

Subverted inputs. For a function $f \colon \mathcal{D} \to \mathcal{R}$ agreeing on τ_0, we define

$$\mathsf{subv}_f = \{x \mid x \in \mathcal{D}(\tau_0) \vee \tilde{f}(x) \neq f(x)\},$$

union of the set of all subverted points for the function f and the $\mathcal{D}(\tau_0)$. We consider elements of the domain of τ_0 as subverted points as the outputs of those have no entropy and are hard-coded into an implementation. Thus, we treat all those inputs as subverted points. Clearly, for all function f,

$$|\mathsf{subv}_f| \le q_1 + \epsilon|\mathcal{D}|.$$

where q_1 denotes the size of τ_0. Let $\epsilon_1 := \epsilon + q_1/|\mathcal{D}|$.

Definition 7 (robust point)**.** *Let f agree on τ_0. A point y is called* robust *in f (or the pair (y, f) is called robust) if for all $x \in Q_f^{-1}(y)$,*

$$\Pr_{\beta}\left[x \in \mathsf{subv}_{f_\beta}\right] \le \sqrt{\epsilon_1}$$

where $\beta \leftarrow\$\ \mathcal{R}$ and $f_\beta := f|_{y \to \beta}$.

Note that the robustness of y in f does not depend on the value $f(y)$. In other words, if y is robust in f, then so in $f|_{y\to\beta}$ for all β.

Definition 8 (popular point)**.** *A point $y \notin \mathcal{D}(\tau_0)$ is called* popular *for a function f if $|Q_f^{-1}(y)| > \epsilon_1^{-1/4}$.*

Recall that the subversion algorithm $\tilde{f}$ makes at most $\tilde{q}$ many queries for any y. So, $\sum_y |Q_f^{-1}(y)| \le \tilde{q}|\mathcal{D}|$. Using the simple averaging argument, the number of popular points is at most $\tilde{q}\epsilon_1^{\frac{1}{4}}|\mathcal{D}|$.

$$\Pr_{x,f}[x \text{ is popular in } f] \le \tilde{q}\epsilon_1^{\frac{1}{4}} \tag{5.1}$$

We call the robust pair (y, f) **good** if (1) y is not popular for f and (2) for all $x \in Q_f^{-1}(y)$, $x \notin \mathsf{subv}_f$. In particular, for good (y, f), it holds that $y \notin \mathsf{subv}_f$ and $y \notin \mathsf{subv}_{f_\beta}$ with high probability over randomness of β where $f_\beta := f|_{y\to\beta}$.

Lemma 11. *For a random $y \leftarrow\$\ \mathcal{D}$, we have*

$$\Pr_{y,\mathsf{f}}[(y, \mathsf{f}) \textit{ is not good}] \le 3\tilde{q}\epsilon_1^{\frac{1}{4}}.$$

Proof. We define two indicator functions:

$$d(x, f) = \begin{cases} 1, & \text{if } x \in \mathsf{subv}_f \\ 0, & \text{otherwise} \end{cases} \quad d_{j,\beta}(x, f) = \begin{cases} 1, & \text{if } x \in \mathsf{subv}_{f|_{\gamma_j^{(x)}\to\beta}} \\ 0, & \text{otherwise.} \end{cases}$$

In other words, $d(x, f)$ is simply an indicator function for capturing crooked points, and $d_{j,\beta}(x, f)$ is an indicator function capturing whether a point x becomes crooked for f after replacing the j-th query output with β. For $1 \le j \le \tilde{q}$, let $D^j(x, f) = \mathsf{Ex}_\beta(d_{j,\beta}(x, f))$. For any function $g \in \Gamma_{\tau_0}$, let $\mathcal{S}_{x,g} := \{(f, \beta) : f|_{\gamma_j^{(x)}\to\beta} = g\}$. It is easy to see that we have $|\mathcal{S}_{x,g}| = 2^n$. Now, for each j,

$$\begin{aligned} \mathsf{Ex}_{x,\mathsf{f}}\left(D^j(x, \mathsf{f})\right) =& \mathsf{Ex}_{x,\mathsf{f}}\mathsf{Ex}_\beta\left(d_{j,\beta}(x, \mathsf{f})\right) \\ =& \sum_{x,\mathsf{f},\beta} \Pr(\mathsf{f}) \Pr(x) \Pr(\beta) \cdot d_{j,\beta}(x, \mathsf{f}) \end{aligned}$$

$$
\begin{aligned}
&= 2^{-n} \sum_{(f,\beta)\in\mathcal{S}_{x,g}} \sum_{x,g} \Pr(g)\Pr(x)\cdot d(x,g) \\
&= \sum_{x,g} \Pr(g)\Pr(x)\cdot d(x,g) \\
&= \mathsf{Ex}_{x,g} d(x,g) \leq \epsilon + \frac{q_1}{|\mathcal{D}|} := \epsilon_1
\end{aligned}
$$

Applying Markov inequality, we get for every $j \in (\tilde{q}]$

$$
\Pr_{x,\mathsf{f}}\left[D^j(x,\mathsf{f}) \geq \epsilon_1^{\frac{1}{2}}\right] \leq \frac{\mathsf{Ex}_{x,\mathsf{f}}\big(D^j(x,\mathsf{f})\big)}{\epsilon^{\frac{1}{2}}} \leq \epsilon_1^{\frac{1}{2}} \tag{5.2}
$$

We recall there are three ways x can be not good in f.

$$
\begin{aligned}
&\Pr_{\mathsf{f},x}[(x,\mathsf{f}) \text{ is not good}] \leq \Pr_{\mathsf{f},x}[x \text{ is popular for } \mathsf{f}] + \\
&\Pr_{\mathsf{f},x}[x \text{ is queried by some point in } \mathsf{subv}_\mathsf{f}] + \\
&\Pr_{\mathsf{f},x}[(x,\mathsf{f}) \text{ is not robust} \mid x \text{ is not popular for } \mathsf{f}]
\end{aligned}
$$

As there are at most $\epsilon_1|\mathcal{D}|$ many points in subv_f,

$$
\Pr_{\mathsf{f},x}[x \text{ is queried by some point in } \mathsf{subv}_\mathsf{f}] \leq \tilde{q}\epsilon_1.
$$

From the definition of robust points and Equation **??**

$$
\begin{aligned}
\Pr_{x,\mathsf{f}}[x \text{ is non robust in } \mathsf{f} \mid x \text{ is not popular for } \mathsf{f}] &\leq \epsilon_1^{-1/4} \sum_{j=1}^{\tilde{q}} \Pr_{x,\mathsf{f}}\left[D^j(x,\mathsf{f}) \geq \epsilon_1^{\frac{1}{2}}\right] \\
&\leq \tilde{q}\epsilon_1^{\frac{1}{4}}
\end{aligned}
$$

Adding above two inequalities and Equation 5.1

$$
\Pr_{\mathsf{f},x}[x \text{ is not good in } \mathsf{f}] \leq \tilde{q}\left(\epsilon_1 + \epsilon_1^{\frac{1}{4}} + \epsilon_1^{\frac{1}{4}}\right) \leq 3\tilde{q}\epsilon_1^{\frac{1}{4}}
$$

□

Critical Set. We consider a set $\mathcal{G}$ of pairs (R,f) of initial values R and functions f satisfying the condition that for every $m \in \{0,1\}^n$ there exists $1 \leq i \leq l$ such that $(\alpha_i := (i, m \oplus R_i), f)$ is good. The following lemma says that for a uniformly random string R (initial value) and a randomly chosen function f agreeing on τ_0, with high probability (R,f) is in the critical set.

Lemma 12. *Let $\tilde{q} \leq 2^{n/2}, \epsilon_1 \leq \frac{1}{2^{16}}$ and $\ell > 2n$. It holds that*

$$
\Pr_{R,\mathsf{f}}((R,\mathsf{f}) \notin \mathcal{G}) \leq 3\tilde{q}\epsilon_1^{1/8} + 2^{-n}.
$$

Proof. We know that $\Pr_\mathsf{f}\left[\Pr_x[(x,\mathsf{f}) \text{ is not good}] > \epsilon_1^{1/8}\right] \leq 3\tilde{q}\epsilon_1^{1/8}$. We say f is convenient if $\Pr_x[(x,f) \text{ is not good}] \leq \epsilon_1^{1/8}$. Fix a convenient f

$$
\Pr_R[(R,f) \notin \mathcal{G}]
$$

$$\leq \sum_{m} \prod_{i=1}^{l} \left(\Pr_{R_i}[(i, m \oplus R_i) \text{ is not good in } f] \right)$$
$$\leq 2^n \times \left(\epsilon_1^{1/8}\right)^l \leq 1/2^n.$$

In the first step, the sum is taken over $m \in \{0,1\}^n$. The last inequality follows from $l > n$, and $\epsilon_1 \leq \frac{1}{2^{16}}$. Hence, we have

$$\Pr_{R,\mathsf{f}}((R, \mathsf{f}) \notin \mathcal{G}) \leq \Pr_{\mathsf{f}}[\mathsf{f} \text{ is not convenient}] + \Pr_{R}[(R, \mathsf{f}) \notin \mathcal{G} | \mathsf{f} \text{ is convenient}]$$
$$\leq 3\tilde{q}\epsilon_1^{1/8} + 1/2^n. \qquad .$$

5.6 Crooked Indifferentiability of Enveloped XOR Construction

In this section, we analyse the crooked indifferentiability security of the EXOR construction. Our main result in this section is Theorem 3.

Theorem 3. *Let $l = 3n + 1, \tilde{q} \leq 2^{n/2}$ and $\epsilon_1 = \epsilon + \frac{q_1}{(l+1)2^n} \leq \frac{1}{16}$. Let $f : [l] \times \{0,1\}^n \to \{0,1\}^n$ be a family of random functions and $\mathsf{EXor} : \{0,1\}^n \to \{0,1\}^n$ be the enveloped-xor construction. Then there exists a simulator S such that for all $((q_1, \tilde{q}), (q_2, q_{sim}), \epsilon, \delta)$ crooked distinguisher $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$*

$$\mathbf{Adv}^{\mathsf{crooked\text{-}indiff}}_{\mathcal{A},(\mathsf{EXor},f)} \leq (4l^2\tilde{q})q_2^2/2^n + (4\tilde{q} + 2l)q_2\epsilon_1^{1/16}$$

The simulator is described in Fig 5.1, which makes at most q_2 query to the random oracle $\mathcal{F}$ and makes $q_2 l\tilde{q}$ many calls to the subverted implementation $\tilde{f}$.

Proof. We recall that, in the real world, the distinguisher is interacting with the subverted construction $\widetilde{\mathsf{EXor}}$ which is defined as

$$\widetilde{\mathsf{EXor}}(R, m) = \tilde{f}(0, \tilde{g}_R(m)) \quad \text{where} \quad \tilde{g}_R(m) = \bigoplus_{i=1}^{l} \tilde{f}(i, m \oplus r_i).$$

We also define a hybrid construction $\overline{\mathsf{EXor}}[f](R, m) = f(0, \tilde{g}_R(m))$. Now consider an adversary $\mathcal{A}$ interacting with $(\mathsf{f}, \overline{\mathsf{EXor}} := \overline{\mathsf{EXor}}[\mathsf{f}])$ in the second phase.

Bad Events

We consider the bad event happening immediately after the i-th query of the adversary, which is of the form (j, x_i) for $j > 0$. We write $m_i = x_i + R_j$. We define four bad events.

1. $\mathsf{bad1}_i$ holds if $(0, \tilde{g}_R(m_i)) \in \mathsf{subv}_f$
2. $\mathsf{bad2}a_i$ holds if $(0, \tilde{g}_R(m_i)) \in \mathcal{D}(\tau_{i-1})$
3. $\mathsf{bad2}b_i$ holds if $\tilde{g}_R(m_i) = \tilde{g}(m_j)$ for some $j < i$
4. $\mathsf{bad2}c_i$ holds if $(0, \tilde{g}_R(m_i)) \in Q(x)$ for some $x \in \mathcal{D}(\tau_i)$ and $x \in \mathsf{subv}_f$.

Let $\mathsf{bad1} = \vee_i \mathsf{bad1}_i$, $\mathsf{bad2} = \vee_i(\mathsf{bad2}a_i \vee \mathsf{bad2}b_i \vee \mathsf{bad2}c_i)$, and $\mathsf{bad} = \mathsf{bad1} \vee \mathsf{bad2}$.

```
$\mathcal{O}(j,x)$
if $(j,x,z) \in L_f$   return $z$
$z \leftarrow\$ \{0,1\}^n$
Add the entry $(j,x,z) \to L_f$
return $z$
```

```
$\tilde{g}_R(M)$
$Sum = 0^n$
for $j = 1$ to $\ell$ do
    $Sum = Sum \oplus \tilde{\mathcal{O}}(j, M \oplus R_j)$
endfor
return $Sum$
```

```
offline phase
    for all $(i, M_k \oplus R_i) \in L^A$
        recompute $\tilde{g}_R(M_k)$ and update $L_f$
```

```
$\tilde{\mathcal{O}}(j,x)$ $(j > 0)$
return $\tilde{h}^{\mathcal{O}}(j,x)$
```

```
Main$(j,x)$
if $j = 0$
    $temp = \mathcal{O}(0,x), L^A = L^A \cup \{(0,x,temp)\}$,
    return $temp$
$M = x \oplus R_j, L_M = \emptyset, S = \tilde{g}_R(M)$
if $(0,S,t) \in L^A$   $\mathsf{bad2}a = 1$
else  Add $(0,S,\mathcal{F}(M))$ to $L$
if $(0,S,z) \in L_f$
    Overwrite the entry$(0,S,\mathcal{F}(M))$
for $i = 1$ to $\ell$
    Add $(i, \mathcal{O}(i, M \oplus R_i))$ to $L_M$
return $L_M$
```

Figure 5.1: Simulator for EXor: Offline Phase is executed after all the distinguisher queries.

Claim 2.

$$\Delta_{\mathcal{A}_2(r,\tilde{\tau},R)}\big((f,\overline{\mathsf{EXor}}(R,\cdot))\ ;\ (f,\widetilde{\overline{\mathsf{EXor}}}(R,\cdot))\big) \le \Pr(\mathsf{bad1})$$

where $\mathsf{bad1}$ *holds while* $\mathcal{A}$ *interacting with* $(\mathsf{f},\overline{\mathsf{EXor}})$.

Proof of the above claim is straightforward as both worlds behave identically until $\mathsf{bad1}$ does not hold.

We have defined our simulator $S^{\mathcal{F}}$ in Figure 5.1 where $\mathcal{F} : \{0,1\}^n \to \{0,1\}^n$ is a random function. The simulator has also observed the above bad events, in particular, $\mathsf{bad2}$. Now we claim that the hybrid construction and the ideal world are indistinguishable provided $\mathsf{bad2}$ does not hold (in the hybrid world) while $\mathcal{A}$ interacting with $(\mathsf{f},\overline{\mathsf{EXor}})$.

Claim 3.

$$\Delta_{\mathcal{A}_2(r,\tilde{\tau},R)}\big((f,\overline{\mathsf{EXor}}(R,\cdot))\ ;\ (S^{\mathcal{F},\tilde{f}}(\tilde{\tau},R),\mathcal{F})\big) \le \Pr(\mathsf{bad2}).$$

We call a transcript good if $\mathsf{bad2}$ does not hold. In the case of the simulator world, whenever $\mathsf{bad2}$ does not hold, the simulator maintains an extended transcript consistent with the hybrid world. As the simulator sets all outputs of the function either randomly or through outputs of $\mathcal{F}$, realising any such good transcript τ' has probability $2^{-n\sigma}$ where $\sigma = |\tau' \setminus \tau_0|$. We have already seen that the probability of realising a good transcript in the hybrid world is exactly $2^{-n\sigma}$. In other words, both worlds behave identically until $\mathsf{bad2}$ does not hold. Combining Claims 2 and 3, we get

$$\mathbf{Adv}^{\mathsf{crooked\text{-}indiff}}_{\mathcal{A},(\mathsf{EXor},f)} \le \Pr[\mathsf{bad}].$$

The proof of Theorem 3 follows from the following lemma. □

Lemma 13.

$$\Pr[\mathsf{bad}] \le (4l^2\tilde{q})q_2^2/2^n + (4\tilde{q} + 2l)q_2\epsilon_1^{1/16}$$

The lemma is proved in section 5.7.

5.7 Proof of Lemma 13

We write $f \Rightarrow_j \tau_j$ to denote the event that after j queries to $(f, \overline{\mathsf{EXor}})$, an adversary obtains the transcript τ_j. We skip the notation j if it is understood from the context.

Definition 9. *A transcript τ_{i-1} is good if*

$$\Pr((R, \mathsf{f}) \in \mathcal{G} \mid \mathsf{f} \Rightarrow \tau_{i-1}) \geq 1 - 3\tilde{q}\epsilon_1^{1/16}.$$

Applying Markov inequality on Lemma 12, we have $\Pr(\tau_{i-1} \text{ is good}) \geq 1 - \epsilon_1^{1/16}$. Let us fix a good transcript τ_{i-1} (which also determines m_i for the i-th query) and a function f agreeing on τ_{i-1} such that $(R, f) \in \mathcal{G}$.

Definition 10. *For any fix k, we say that f is called τ_i-good if (i) $f \Rightarrow \tau_{i-1}$ and (ii) (α_k, f) is good.*

Claim 4. *For any τ_i-good f there exists a set S of size at least $2^n(1 - \epsilon_1^{1/4})$ such that for all $\beta \in S$, $f_\beta := f|_{\alpha \to \beta}$ is also τ_i-good.*

Proof. We fix a function $f \in \Gamma_{R,\tau_{i-1},\pi_{i-1}}$ such that (α_k, f) is τ_i-good. Now we identify a set of good values of β such that $f_\beta := f|_{\alpha_k \to \beta} \in \Gamma_{R,\tau_{i-1},\pi_{i-1}}$ such that (α_k, f) is τ_i-good. In other words, setting the output of f on the point α_k to β keeps the pair (α_k, f_β) good. For every $x \in \mathcal{D}(\tau_{i-1}) \cap Q_f^{-1}(\alpha_k)$, let B_x denote the set of all bad β values for which good condition of (α_k, f) gets violated. By definition, $|B_x| \leq \epsilon_1^{1/2}$ and hence $| \cup_x B_x| \leq 2^n \epsilon_1^{1/4}$. We define

$$S = \mathcal{D} \setminus \cup_{x \in Q_f^{-1}(\alpha_k)} B_x.$$

Note that for all $\beta \in S$, (α_k, f_β) is τ_i-good.

□

Due to the above claim, we have

$$\Pr(\mathsf{f}(\alpha_k) = z \mid (\alpha_k, \mathsf{f}) \text{ is } \tau_i\text{-good, }) \leq \frac{1}{|S|} \leq \frac{1}{2^n(1 - \epsilon_1^{1/4})} \leq \frac{2}{2^n}.$$

The last inequality holds because $\epsilon_1 \leq \frac{1}{16}$. Now note that for any event E, we have

$$\begin{aligned}
\Pr(E|\tau_{i-1}) &\leq \Pr_{\mathsf{f}}(E \wedge (R, \mathsf{f}) \in \mathcal{G}|\tau_{i-1}) + 3\tilde{q}\epsilon_1^{1/16} \\
&\leq \sum_{k=1}^{l} \Pr_{\mathsf{f}}(E \wedge (\alpha_k, \mathsf{f}) \text{ is } \tau_{i-1}\text{-good} \mid \tau_{i-1}) + 3\tilde{q}\epsilon_1^{1/16} \\
&\leq \sum_{k=1}^{l} \Pr_{\mathsf{f}}(E \mid (\alpha_k, \mathsf{f}) \text{ is } \tau_{i-1}\text{-good }) + 3\tilde{q}\epsilon_1^{1/16}
\end{aligned}$$

For the last inequality, we simply use the fact that

$$\Pr_{\mathsf{f}}((\alpha_k, \mathsf{f}) \text{ is } \tau_{i-1}\text{-good} \mid \tau_{i-1}) \leq 1.$$

Now we bound each bad event, and then we can multiply by l and add all the terms to get the bound.

Bound of $\Pr(\mathsf{bad2}a_i \cup \mathsf{bad2}b_i)$

Fix a τ_i-good f. Let $B2$ denote the set of all elements containing $\tilde{g}_R(m_j)$ (for all $j < i$) and all elements from $\mathcal{D}(\tau_{i-1})$ of the form $(0, *)$. Note that the set $B2$ and $\sum_{j \neq k} \tilde{f}(m_i + R_j)$ does not depend on the value $f(\alpha_k)$ provided $f(\alpha_k) \in S$. Hence,

$$\Pr_{\mathsf{f}}(\mathsf{bad2}a_i \cup \mathsf{bad2}b_i \mid (\alpha_k, \mathsf{f}) \text{ is } \tau_{i-1}\text{-good }) \leq 2i/2^n.$$

Bound of $\Pr(\mathsf{bad2}c_i)$

We first note that for all $\beta \in S$ and an input x which queries α_k, x is not crooked and a robust point. Let $A = \mathcal{D}(\tau_i) \setminus (\{\alpha_k\} \cup Q_f^{-1}(\alpha_k))$. Let $\tilde{A}$ denote the set of all points queried by the elements of A. Suppose $\tilde{g}_R(m_i) \notin \tilde{A}$. Then, for every x from the domain of τ_i querying $\tilde{g}_R(m_i)$ must query α_k and hence $\mathsf{bad2}c_i$ does not hold. So, $\mathsf{bad2}c_i$ can hold only if $\tilde{g}_R(m_i) \in \tilde{A}$. Once again by randomness of $f(\alpha_k)$, we have

$$\Pr(\mathsf{bad2}c_i \mid (\alpha_k, \mathsf{f}) \text{ is } \tau_{i-1}\text{-good }) \leq 2\tilde{q}il/2^n.$$

Bound of $\Pr(\mathsf{bad1}_i)$

Clearly, $\tilde{f}_\beta(x)$ can be different from $\tilde{f}(x)$, only if $x \in Q_f^{-1}(\alpha_k)$. Moreover for every $x \in Q_f^{-1}(\alpha_k)$, as both (α_k, f) and (α_k, f_β) are good, it holds that $x \notin \mathsf{subv}_f$ and $x \notin \mathsf{subv}_{f_b}$. Thus for any such τ_i-good f, f_β, we have the following conditions: $\mathsf{subv}_f = \mathsf{subv}_{f_\beta}$. Thus,

$$\Pr[\mathsf{bad1}_i \mid (\alpha_k, f) \text{ good}, \tau_i \text{ good}] \leq 2\epsilon_1$$

So,

$$\Pr[\mathsf{bad}_i \mid \tau_{i-1}] \leq 4l^2 q_2 \tilde{q}/2^n + 2l\epsilon_1 + 3\tilde{q}\epsilon_1^{1/16}$$

Finally, we add the probability that we realise a not good transcript τ_{i-1} and we obtain bound for $\Pr(\mathsf{bad}_i)$. By taking union bound over $i \in [q_2]$, we get

$$\begin{aligned}\Pr[\mathsf{bad}] &\leq 4l^2 q_2^2 \tilde{q}/2^n + 2lq_2\epsilon_1 + 3\tilde{q}q_2\epsilon_1^{1/16} + q_2\epsilon_1^{1/16} \\ &\leq (4l^2\tilde{q})q_2^2/2^n + (4\tilde{q} + 2l)q_2\epsilon_1^{1/16}\end{aligned}$$

This finishes the proof of Lemma 13. □

Chapter 6

Subversion Resilient Hashing: Efficient Constructions and Modular Proofs for Crooked Indifferentiability

6.1 Introduction

We consider the problem of constructing secure cryptographic hash functions from *subverted* ideal primitives. Hash functions are used to instantiate Random Oracles in cryptographic protocols. The notion of indifferentiability security is a popular tool for certifying the structural soundness of a hash design for such instantiations. In CRYPTO 2018, Russell, Tang, Yung, and Zhou introduced the notion of crooked indifferentiability to extend this paradigm even when the underlying primitive of the hashing mode is subverted. They showed that an n-to-n-bit function implemented using Enveloped XOR construction (EXor) with $3n+1$ many independent n-to-n-bit functions and $3n^2$-bit random seed can be proven secure asymptotically in the crooked indifferentiability setting. Unfortunately, known techniques to prove crooked indifferentiability are extremely complicated, and no practical hashing mode has been analysed in this setting.

- We introduce new techniques to prove crooked indifferentiability. We establish that upper bounding the subversion probability of a chaining query is sufficient to argue subversion resistance of a standard indifferentiable mode of operation. Our technique links standard indifferentiability and crooked indifferentiability and circumvents the complications of proving the consistency of the simulator in the crooked setting.
- We prove the crooked indifferentiability of the Sponge construction when the underlying primitive is modelled as an n-to-n-bit random function. Our proofs only require n-bit randomly chosen but fixed IV and do not mandate any independent function requirement. The result naturally extends to the Merkle-Damgård domain extension with prefix-free padding. Our results minimise required randomness and solve the main open problem raised by Russell, Tang, Yung, and Zhou.

Traditionally, cryptographic hash functions are designed by applying a domain extension algorithm on suitable primitives of a smaller domain. Security of the hash functions is often derived via information-theoretic arguments, assuming the underlying primitives behave as ideal where the adversary is permitted only to query the primitives. In practice, however, the implementations of the primitives may leak more information to the adversary and possibly even allow malicious tampering. A good example is the Dual-EC tampering attack [29] which led to the withdrawal of a standardised PRG due to a potential backdoor in the implementation.

The framework of Kleptography, introduced by Young and Yung [100, 101] more than twenty years ago, allows a "proud but curious" adversary to replace a cryptographic implementation with a crooked version intending to subvert its security without getting caught. Bellare, Paterson, and Rogaway [9] revitalised the framework under the name of Algorithmic Substitution Attack (ASA). They showed that it is possible to mount an algorithm substitution attack against almost all known symmetric key encryption schemes to the extent that the attacker learns the secret key. A series of work has been done in recent years formalizing approaches to resist algorithm substitution attacks [48, 8, 70, 44, 45, 88, 89].

The notion of crooked indifferentiability from a Random Oracle and the composition theorem proved in [90] guarantees that a construction proved secure in this framework can be used to replace a Random Oracle in any single-stage game in the kleptographic setting. While popular hash functions are the most natural choice for instantiating the Random Oracle, their suitability is still unknown. We ask, *can the popular hashing modes, for some parameters, achieve this many-fold stronger security notion?* Given the surge of new constructions in the ASA setting [32, 3, 4], the importance of the question cannot be overstated.

Proving a construction secure in the crooked indifferentiability setting is an immensely challenging task. Unlike the classical setting where the adversary is passive, the crooked indifferentiability adversary is active and could subvert any algorithm. The only known crooked indifferentiability bound is for the construction called Enveloped XOR (EXor), depicted in Figure 6.1. In [90], the authors using the rejection-sampling technique showed the security of EXor construction. The instantiation requires $3n + 1$ many independent functions and n^2 many random bits. We note, however, that the Enveloped XOR construction produces an n-bit to n-bit random function. Instantiating a hash function would require applying domain extension techniques on top of it, implying more function calls and possibly more independent random bits. *Minimising randomness and reducing the number of function calls while still achieving crooked indifferentiability was left as the main challenge in [90].*

Finally, the technique of [90], though ingenious, is very complex. It is difficult to give an intuitive justification for why the construction and the approach work. The alternative proof of [20] is also quite involved. Given that we have established tools to prove indifferentiability in the classical setting, it is natural to ask whether we can leverage those tools to prove crooked indifferentiability.

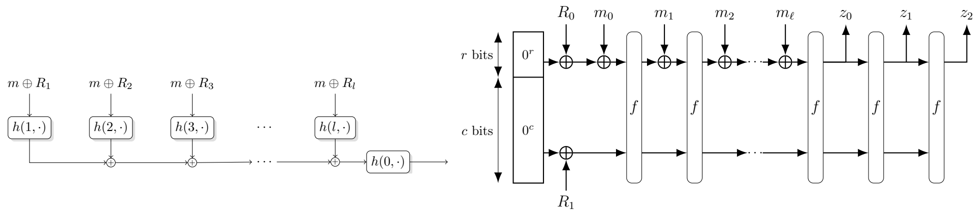


Figure 6.1: EXor construction (left) and Sponge Construction with random IV (right).

6.2 Our Contributions

We introduce new techniques to prove crooked indifferentiability and establish security bounds for popular hashing modes, the Sponge construction, and the ubiquitous Merkle-Damgård construction. We elaborate on our contributions below.

New Techniques for Crooked Indifferentiability. We present new techniques to prove crooked indifferentiability. We introduce a new security game called Force-Crook, where the challenge to the adversary is to produce a message for which the construction makes a primitive query on a subverted input. We show that bounding the advantage of the adversary in the Force-Crook game is sufficient to prove the crooked indifferentiability of constructions which are secure under the classical indifferentiability paradigm.

Crooked Indifferentiability of Popular Hashing Modes. We apply our techniques to prove the security of popular hashing modes. Our main contribution is to show that the Sponge construction, instantiated with a random function and a randomised initial value, is crooked-indifferentiable from a Random Oracle. The construction uses the same function at every iteration. The design is identical to the one proven indifferentiable in [13]. This result positively answers our quest for a practical crooked-indifferentiable hashing mode. Moreover, the proof requires only a linear (in terms of the security parameter) number of random bits and thus answers the *main open question* raised by RTYZ [90].

We show that the technique with a minor modification is sufficient to prove the security of the classical Merkle-Damgård construction with prefix-free padding. The hash function uses an $n+1$-to-n-bit compression function.

6.2.1 Overview of Our Techniques

Technical Challenges in Crooked Indifferentiability.

The main challenge in the crooked setting is to prove the randomness of the construction's output. As the underlying primitives are subverted, the adversary may have full information about the function on some points without querying the oracles. Consider the following example. We are given an n-to-n-bit random function f. By definition, f is classically indifferentiable from a random oracle. Now consider a simple subverted implementation $\tilde{f}$ of f. The program $\tilde{f}$ honestly implements f everywhere except at point 0, where it outputs $\tilde{f}(0) = 0$. Such an $\tilde{f}$ can be easily distinguished from a random oracle.

The established technique to correct the situation would be the random-masking technique, but that does not work either. Consider, for example, simple input masking with a random string R obtained by the function $g_R(M) \overset{\text{def}}{=} f(MR)$. As the string R is fixed at the start of the game (after the adversary submits the subverted implementation), the distinguisher can indeed choose the message $M = R$, resulting in a distinguishing condition $g_R(R) = 0$. From the above two examples, one can abstract out the first challenge of proving crooked indifferentiability. *The output distribution of the underlying primitive, conditioned on the adversary's view, is not uniform for every point.* The challenge becomes even more daunting when we consider an implementation that can subvert a point based on the function evaluations at that and possibly some other points. We can no longer assume function values are independently distributed. Thus the tools and techniques developed for classical indifferentiability seem to be useless here.

The Intermediate Game Force-Crook.

We found a seemingly obvious but powerful technique to handle subversions. The difference between the real world in the crooked indifferentiability and the real world in the classical indifferentiability setting is only in the oracle of the construction C. In the crooked setting, C is given oracle access to $\tilde{f}$ whereas, in the classical setting, C queries the primitive f itself. As long as no chaining value results in querying f on a crooked point, the output distributions of these two worlds are identical! In other words, if for every message M submitted by the adversary to C, it holds with a high probability that $C^f(M) = C^{\tilde{f}}(M)$, then (C^f, f) and $(C^{\tilde{f}}, f)$ are indistinguishable. If C is indifferentiable in the classical setting, then that simulator would work perfectly as the simulator in the crooked setting.

In Section 6.4, we introduce a security game Force-Crook where the adversary is challenged to find a message where $C^f(M) \neq C^{\tilde{f}}(M)$. We show that for a construction which is proven indifferentiable from a random oracle in the classical setting (with security bound δ_i), the crooked indifferentiability advantage is bounded by the advantage of winning the Force-Crook game plus δ_i.

Bounding Winning Advantage of Force-Crook

To bound the adversary's success probability of winning the game Force-Crook, we focus on ensuring all the chaining inputs remain uncrooked with high probability. Our intuition is to argue that if a chaining query is uncrooked, the output is uniform. Given that only a negligible fraction of points are crooked, when we use random iv, the first chaining inputs are random and, thus, with high probability, uncrooked. Suppose only a few bits of the message are injected at every iteration. Then, the following chaining query input is close to being uniform, and thus, with high probability, uncrooked as well. Now we can repeat this argument throughout the computation of C. For the Sponge and Merkle-Damgård constructions, this idea in itself is sufficient for handling simple subversion.

We explain it in more detail for the following simplified setting. Suppose the subverted implementation $\tilde{f}$ is such that on input a point x, the output of $\tilde{f}(x)$ depends only on the value of $f(x)$, and it is independent of $f(y)$ for all $y \neq x$. Consider the Sponge construction based on a random function $f : \{0,1\}^n \to \{0,1\}^n$. By definition of worst-case subversion by a proud but curious adversary, for *all* choices for the function f, at most ϵ fraction of the inputs are crooked ($\tilde{f}(x) \neq f(x)$). In addition, there are at most q_1 many points queried by the implementor before producing the subverted implementation. Hence for every function f, there is a set S_f of size at least $(1-\epsilon)2^n - q_1$ whose members are neither fixed by the implementor nor subverted. For a randomly chosen function f and a random string with overwhelming probability, the random string will be a member of S_f. If we set the rate part of the Sponge construction to be 1, for both the choice of $m_0 \in \{0,1\}$, the first chaining query to f will be a member of S_f with probability $(1 - 2\epsilon - \frac{2q_1}{2^n})$.

We can repeat the above argument inductively. Consider the lazy sampling framework of random functions. We say a chaining query x_i is good if, for all choices of $m_{i+1} \in \{0,1\}$, the next chaining query $x_{i+1} = f(x_i)m_{i+1}$ is subverted with low probability (say $\epsilon^{\frac{1}{2}}$). In other words, x_{i+1} is a member of S_f with high probability. One can show that a randomly chosen point is good with high probability. As $f(x_i)$ is uniformly distributed, x_{i+1} would also be a good chaining query. For the base case of the induction argument, we recall that the first chaining query is generated from the initial random string. For all values of $m_0 \in \{0,1\}$, it is a good chaining query with high probability. Thus, we get all the chaining queries would be good, and by extension, all the chaining queries will be uncrooked with overwhelming probability.

The matter gets complicated when we consider a general $\tilde{f}$ whose output can depend on adaptively chosen multiple points. With careful analysis, *we extend our arguments to this general case.* In Section 6.5, we present the analysis in detail.

6.2.2 Impact of Our Results

Subversion Agnostic Indifferentiability.

We achieve a strong form of crooked indifferentiability where the simulator is subversion agnostic. When we establish crooked indifferentiability via the Force-Crook game, S does not even need access to subverted implementation $\tilde{f}$. While we show Sponge and Merkle-Damgård attain such security, not all constructions achieve such strong crooked indifferentiability. One notable example is the Enveloped Xor construction, where the simulator must have access to $\tilde{f}$ to achieve crooked indifferentiability as formulated in [90]. Thus, our modular proof technique illustrates a simple condition for a classical indifferentiable construction to achieve crooked indifferentiability.

Crooked vs Classical.

A learned reader may observe that a crooked-indifferentiable construction's efficiency and security parameters are worse than what can be proven in the classical indifferentiability setting. One can wonder about the crooked indifferentiability framework's significance and our results' impact. In particular, for the Sponge construction with n bit function, we prove crooked indifferentiability security of asymptotically $n/4$ bits when at each round, *one* bit of message is injected and $\epsilon \leq 1/2^{n/2}$. In contrast, SHA3, with each iteration consuming r bits of messages, achieves $(n-r)/2$ bits of security in the classical indifferentiability setting.

However, comparing bit-security without considering the adversary's power leads to misleading impressions. While proving indifferentiability, we aim to achieve independent and uniformly sampled hash output for *every* point. The classical indifferentiability assumes that an adversary is passive and is content with only black-box access to the underlying primitive. Thus, the primitive could be modelled as ideal. In particular, each point is mapped independently following a high-entropy probability distribution.

In comparison, the adversary in the kleptographic setting is *active*. The implementation of the primitive is subverted. The points are not mapped independently and for some "small" yet non-zero fraction of the inputs, the adversary has carefully *chosen* the function. We can no longer directly leverage the randomness of the underlying primitive. Naturally, the security-efficiency tradeoff achieved in the crooked setting against such an active adversary is somewhat weaker than what is accomplished against the passive adversary of the classical indifferentiability paradigm.

6.3 Suitable Functions and Sets

Let $f\colon \mathcal{D}_f \to \mathcal{R}_f$ be a function. For a transcript τ, we define $C_{f,\tau}$ to be the union of the set of subverted points for the function f and the points fixed by τ.

Definition 11. $C_{f,\tau} = \{x \mid x \in \mathcal{D}(\tau) \vee \tilde{f}(x) \neq f(x)\}$.

By the definition of ϵ-crooked,

$$\frac{|C_{f,\tau}|}{|\mathcal{D}_f|} \leq \epsilon_\tau := \epsilon + \frac{|\tau|}{|\mathcal{D}_f|}.$$

At the beginning of the second stage of the crooked indifferentiability game, the transcript contains the interaction of the q_1 many queries made by the implementer. We define

$$\epsilon_1 = \epsilon + \frac{q_1}{2^n}.$$

Let τ be a (partial) transcript. Recall that we say a function g agrees on a transcript τ when the transcript holds for the function g.

$$\mathsf{F}_{n,n|\tau} \overset{\text{def}}{=} \{g \in \mathsf{F}_{n,n} \mid g \text{ agrees on } \tau\}.$$

6.4 From Classical Indifferentiability to Crooked Indifferentiability

In this section, we establish sufficient conditions to lift the classical indifferentiability results to the crooked indifferentiability setting.

Security Games. The results in this work are proven in the framework of code-based games [10]. A game G consists of a `main` oracle and zero or more stateful oracles $O_1, O_2, \ldots, O_n$. If a game G is implemented using a function f, we write $G[f]$ to denote the game. The success probability of algorithm $\mathcal{A}$ in the game G is defined by $\mathbf{Succ}_{\mathcal{A},G} \overset{\text{def}}{=} \Pr\big[G^{\mathcal{A}} = 1\big]$. The query complexity of $\mathcal{A}$ is the number of queries made by $\mathcal{A}$ to its oracles.

Let $f\colon \mathcal{D}_f \to \mathcal{R}$ and $\mathcal{F}\colon \mathcal{D} \to \mathcal{R}$ be two random oracles where $\mathcal{D} \supseteq \mathcal{D}_f$. Let C^f be an $\mathcal{F}$-compatible construction. We consider a crooked distinguisher $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$.

6.4.1 Force-Crook Game

```
Game Force-Crook($C$)
$f \leftarrow\!\!\$\ \mathcal{F}_{n,n}$
$(\tilde{\tau}, \langle\tilde{f}\rangle) \leftarrow \mathcal{A}_1^f$
$M \leftarrow \mathcal{A}_2^{(C^f(\cdot,R),f)}(\tilde{\tau}, R)$
if $C^f(M) \neq C^{\tilde{f}}(M)$
  return 1
else
  return 0
```

Figure 6.2: The Force-Crook game

In this section, we introduce the security game Force-Crook. Formally the game is defined in Figure 6.2. The force-crook advantage of an adversary is defined as

$$\mathbf{Adv}^{\textsf{force-crook}}_{\mathcal{A},C} \overset{\text{def}}{=} \mathbf{Succ}_{\mathcal{A},\textsf{force-crook[C]}}.$$

Given a construction C, we define

$$\mathbf{Insec}^{\textsf{force-crook}}_{C,(q_1,\tilde{q},\epsilon),q_2} \overset{\text{def}}{=} \max_{\mathcal{A}} \mathbf{Adv}^{\textsf{force-crook}}_{\mathcal{A},C}.$$

where the maximum is taken over all $((q_1, \tilde{q}, \epsilon), q_2)$-crooked-distinguishers.

6.4.2 Achieving Crooked Indifferentiability

Our main technique to prove the security of Sponge and prefix-free Merkle-Damgård constructions results from Theorem 4. The idea is depicted in Figure 6.3. Suppose C is indifferentiable from $\mathcal{F}$ (the advantage of distinguishing middle and rightmost worlds is small). If the Force-Crook advantage is small, then the advantage of distinguishing between the leftmost and the middle-world is small. Then the classical simulator S successfully acts as the simulator in the real world of the crooked setting.

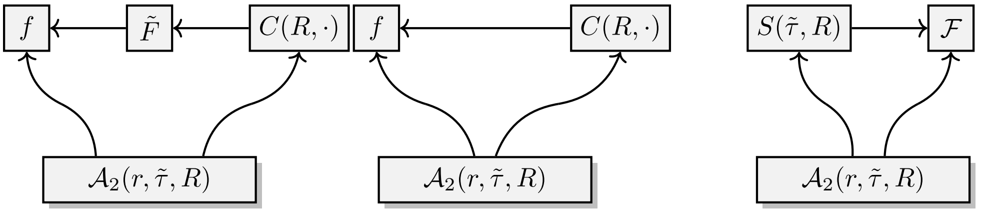

f
$\tilde{F}$
$C(R,\cdot)$
$\mathcal{A}_2(r,\tilde{\tau},R)$
f
$C(R,\cdot)$
$\mathcal{A}_2(r,\tilde{\tau},R)$
$S(\tilde{\tau},R)$
$\mathcal{F}$
$\mathcal{A}_2(r,\tilde{\tau},R)$

Figure 6.3: The hybrid: The leftmost picture is the real world of the crooked setting. The middle picture is the real world in the classical setting. The rightmost picture is the ideal world in the classical setting.

Theorem 4. *Let $C^f : \mathcal{D} \to \mathcal{R}$ be a hash function built on primitive $f : \mathcal{D}_f \to \mathcal{R}$. Let C^f be $((q_P, q_C, q_{\text{sim}}), \delta_i)$-indifferentiable from a random oracle $\mathcal{F}$. C^f is $((q_1, \tilde{q}),\ (q_2, q_{\text{sim}}),\ \epsilon, \delta_c)$-crooked-indifferentiable from $\mathcal{F}$ where*

$$\delta_c \leq \delta_i + \mathbf{Insec}^{\textsf{force-crook}}_{C,(q_1,\tilde{q},\epsilon),q_2}$$

and $q_1 + q_2 \leq q_P$.

Proof. From the definitions and using triangle inequality, we get

$$\delta_c \leq \delta_i + \Delta_{\mathcal{A}_2(r,\tilde{\tau},R)}\big((f, C^{\tilde{f}}(R,\cdot))\ ;\ (f, C^f(R,\cdot))\big).$$

To prove the theorem, we need to show

$$\Delta_{\mathcal{A}_2(r,\tilde{\tau},R)}\big((f, C^{\tilde{f}}(R,\cdot))\ ;\ (f, C^f(R,\cdot))\big) \leq \mathbf{Insec}^{\textsf{force-crook}}_{C,(q_1,\tilde{q}),(q_2,q_s)}.$$

Let $\textsf{bad}$ denote the event $\mathcal{A}_2(r,\tilde{\tau},R)$ makes a query to $C^{\tilde{f}}$ (or C^f) oracle with input M such that

$$C^f(R,M) \neq C^{\tilde{f}}(R,M).$$

Now unless $\textsf{bad}$ is set, the outputs of the oracles in both the world $(f, C^{\tilde{f}}(R,\cdot))$ and $(f, C^f(R,\cdot))$ are the same. Thus we get

$$\Pr[\mathcal{A}_2^{(f,C^{\tilde{f}}(R,\cdot))}(r,\tilde{\tau},R) = 1 \cap \neg\textsf{bad}] = \Pr[\mathcal{A}_2^{(f,C^f(R,\cdot))}(r,\tilde{\tau},R) = 1 \cap \neg\textsf{bad}]. \qquad (6.1)$$

We derive, using Definition **??**, triangle inequality, and Equation 6.1

$$\begin{aligned}
&\Delta_{\mathcal{A}_2(r,\tilde{\tau},R)}\big((f, C^{\tilde{f}}(R,\cdot))\ ;\ (f, C^f(R,\cdot))\big) \\
&= \left|\Pr[\mathcal{A}_2^{(f,C^{\tilde{f}}(R,\cdot))}(r,\tilde{\tau},R) = 1] - \Pr[\mathcal{A}_2^{(f,C^f(R,\cdot))}(r,\tilde{\tau},R) = 1]\right|
\end{aligned}$$

$$\begin{aligned}
&\leq \left|\Pr[\mathcal{A}_2^{(f,C^{\tilde{f}}(R,\cdot))}(r,\tilde{\tau},R) = 1 \cap \mathsf{bad}] - \Pr[\mathcal{A}_2^{(f,C^{f}(R,\cdot))}(r,\tilde{\tau},R) = 1 \cap \mathsf{bad}]\right| + \\
&\qquad \left|\Pr[\mathcal{A}_2^{(f,C^{\tilde{f}}(R,\cdot))}(r,\tilde{\tau},R) = 1 \cap \neg\mathsf{bad}] - \Pr[\mathcal{A}_2^{(f,C^{f}(R,\cdot))}(r,\tilde{\tau},R) = 1 \cap \neg\mathsf{bad}]\right| \\
&= \left|\Pr[\mathcal{A}_2^{(f,C^{\tilde{f}}(R,\cdot))}(r,\tilde{\tau},R) = 1 \cap \mathsf{bad}] - \Pr[\mathcal{A}_2^{(f,C^{f}(R,\cdot))}(r,\tilde{\tau},R) = 1 \cap \mathsf{bad}]\right| \\
&\leq \Pr[\mathsf{bad}].
\end{aligned}$$

The last inequality follows as both $\Pr[\mathcal{A}_2^{(f,C^{\tilde{f}}(R,\cdot))}(r,\tilde{\tau},R) = 1 \cap \mathsf{bad}]$ and $\Pr[\mathcal{A}_2^{(f,C^{f}(R,\cdot))}(r,\tilde{\tau},R) = 1 \cap \mathsf{bad}]$ are numbers between 0 and $\Pr[\mathsf{bad}]$. Finally, if bad happens then $\mathcal{A}_2(r,\tilde{\tau},R)$ wins the game $\mathsf{Force\text{-}Crook}$. Thus

$$\Pr[\mathsf{bad}] \leq \mathbf{Insec}^{\mathsf{force\text{-}crook}}_{C,(q_1,\tilde{q}),(q_2,q_s)}.$$

The theorem follows. □

```
Procedure Sponge
  // random string $R$, Message $m \in \{0,1\}^{\ell}$
  $x = (x_a, x_c) = R$
  for $i = 0$ to $\left\lceil \frac{\ell}{r} \right\rceil - 1$ do
      $(x_a, x_c) = f(x_a \oplus m_i, x_c)$
  endfor
  for $i = 0$ to $\left\lceil \frac{s}{r} \right\rceil - 1$ do
      Append $x_a$ to output
      $(x_a, x_c) = f(x_a, x_c)$
  endfor
```

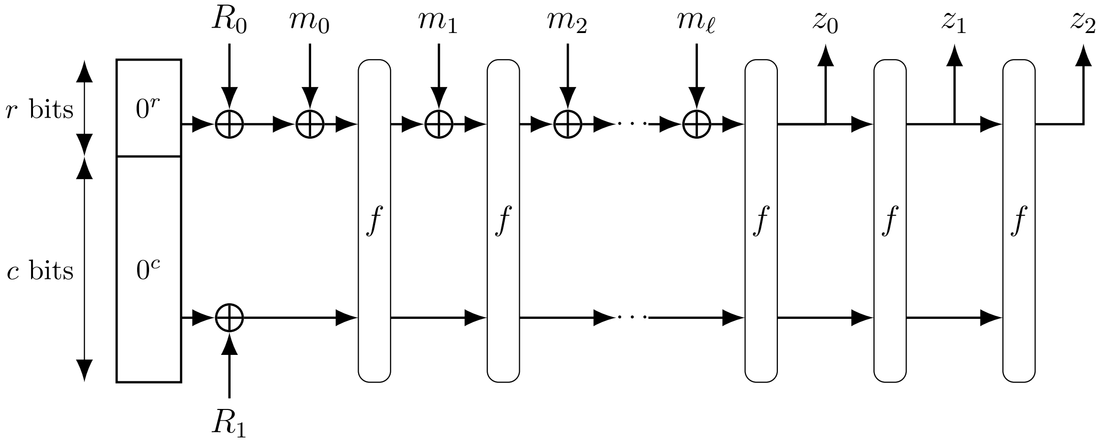


Figure 6.4: Crooked-Indifferentiable Sponge Construction

6.5 Crooked Indifferentiability of Sponge Construction

In this section, we show that the Sponge construction based on an n-to-n-bit random function can be proved crooked-indifferentiable from a random oracle when initialised with a random IV.
Sponge Construction. The details of the parameters of the Sponge construction we consider are listed below.

<u>Target Hash Function.</u> The construction implements a FIL-hash function H such that $H : \{0,1\}^\ell \to \{0,1\}^s$.
<u>Primitives.</u> The underlying primitive of the construction is an n-to-n bit function $f : \{0,1\}^n \to \{0,1\}^n$. In the security proof, f is modelled as a random oracle.
<u>Public Randomness.</u> The public randomness is $R \xleftarrow{\$} \{0,1\}^n$.
<u>Padding.</u> We use the same padding scheme as the original Sponge construction, where it is required that the last message block is non-zero.
<u>Construction.</u> The chaining value of the Sponge construction is divided into two parts, *rate* (length denoted by r) and *capacity* (length denoted by c). The message is divided into r-bit blocks. The construction works in two phases, absorbing and squeezing. In one round of the absorbing phase, one r-bit message block is xored with the rate part of the chaining value. The function f is then applied to the result (of the xor) to get the chaining value of the next round. The construction enters the squeezing phase once all the input message blocks are processed. At each round, the rate part of the chaining value is stored as the output block, followed by the application of f on the whole chaining value. The algorithm stops once we have s bits of output. The construction is described in Figure 6.4.

q_1	Number of f queries made by the implementor $\mathcal{A}_1$
$\tilde{q}$	Number of f queries made by the subverted implementation $\tilde{f}$
q_2	Total number of queries made by the distinguisher $\mathcal{A}_2$
q_{sim}	Total number of $\mathcal{F}$ queries made by the simulator S
ϵ	Fraction of subverted points under $\tilde{f}$

Figure 6.5: Recalling the notations

Our main result in this section is Theorem 5. We recall the notations in Figure 6.5.

Theorem 5. *Let $f : \{0,1\}n \to \{0,1\}n$ be a random function and $C^f : \{0,1\}^\ell \to \{0,1\}^s$ be the Sponge construction. Let r be the rate part, and $c = n - r$ be the capacity part of the chain. Then there exists a simulator S such that for all $((q_1, \tilde{q}, \epsilon), q_2)$-crooked distinguishers $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, it holds that*

$$\mathbf{Adv}^{\textsf{crooked-indiff}}_{\mathcal{A},(C,f)} \leq \mathcal{O}\left(2^r \times \sigma \times \left(\epsilon_1^{\frac{1}{2}} + \frac{\tilde{q}}{2^{\frac{n}{4}}} + \epsilon_1^{\frac{1}{2}} + \frac{\sigma}{2^n} + \frac{\sigma}{2^{\frac{n}{2}}}\right)\right)$$

where $\epsilon_1 = \epsilon + \frac{q_1}{2^n}$, σ is the total number of blocks in the queries made by $\mathcal{A}_2$. The simulator makes $\mathcal{O}(\sigma)$ queries.

The rest of the section is dedicated to proving Theorem 5. First, we recall the result of Bertoni, Daemen, Peeters, and Van Assche [13] to find the classical Indifferentiability bound of the Sponge construction. Then we shall bound the $\mathbf{Insec}^{\textsf{force-crook}}_{C,(q_1,\tilde{q}),(q_2,q_s)}$, the advantage of any distinguisher against our construction in the Force-Crook game. Finally, using Theorem 4, we shall get Theorem 5.
Classical Indifferentiability of Sponge with Random Function. We recall the classical indifferentiability result of Sponge mode from [13] in our notations and parameters.

Theorem 6 (Theorem 1 in [13]). *Let $f : \{0,1\}^n \to \{0,1\}n$ be a random function. The Sponge construction instantiating $C^f : \{0,1\}^\ell \to \{0,1\}^s$ is (q, q_{sim}, δ_i)-indifferentiable from a random oracle for $q_{sim} = \mathcal{O}(\sigma)$ and $\delta_i = \mathcal{O}(\frac{\sigma^2}{2^c})$ where σ is the total number of queries made by the distinguisher.*

We note that in [13], the above theorem is proved to hold for any fixed IV. Thus, we can conclude that the theorem holds for a randomly chosen and then fixed IV, as required in our case.

6.5.1 Bounding Probability of Winning Force-Crook: Sponge on Random Functions

Now we bound $\mathbf{Insec}^{\mathsf{force\text{-}crook}}_{C,(q_1,\tilde{q}),(q_2,q_s)}$. We shall prove the following lemma, which summarises the main findings of this section. We recall the notations in Figure 6.5.

Lemma 14. *Let C be the Sponge construction with randomised IV. Let r be the rate part, and $c = n - r$ be the capacity part of the chain. It holds that*

$$\mathbf{Insec}^{\mathsf{force\text{-}crook}}_{C,(q_1,\tilde{q}),(q_2,q_s)} \leq \mathcal{O}\left(2^r \times \sigma \times \left(\epsilon_1^{\frac{1}{2}} + \frac{\tilde{q}}{2^{\frac{n}{4}}} + \epsilon_1^{\frac{1}{2}} + \frac{\sigma}{2^n} + \frac{\sigma}{2^{\frac{n}{2}}}\right)\right)$$

where $\sigma = q_2(\ell + s) + q_S$.

The Setup of Bounding Adversary's Advantage.

The main idea of our proof is to bound the probability that the adversary can produce a message such that a chaining query is subverted. We need the following definition.

Definition 12 (Robust Point). *A point $x \in \{0,1\}n$ is said to be a (r, ϵ_1)-robust point with respect to a transcript τ, if*

1. *$x \notin \mathcal{D}(\tau)$.*
2. *Define $y_\zeta = f(x)\zeta 0^{n-r}$ for $\zeta \in \{0,1\}^r$. It holds that*

$$\Pr_{f \leftarrow \mathsf{F}_{n,n|\tau}}\left[\bigvee_{\zeta \in \{0,1\}^r} y_\zeta \in C_{f,\tau}\right] \leq 2^r \left(\epsilon_1^{\frac{1}{2}} + \frac{|\tau|}{2^n} + \frac{|\tau|}{2^{\frac{n}{2}}}\right).$$

Popular Points. Consider a point $x \in \mathcal{D}_f \setminus \mathcal{D}(\tau)$. x is called favourite of y with respect to τ if

$$\Pr_{f \leftarrow\$ \mathsf{F}_{n,n|\tau}}[y \twoheadrightarrow_f x] \geq \frac{1}{2^{\frac{n}{2}}}.$$

Definition 13. *x is* popular *with respect to τ if*

$$\Pr_y[x \text{ is favourite of } y] > \frac{1}{2^{\frac{n}{4}}}.$$

Recall that the subversion algorithm $\tilde{f}$ makes at most $\tilde{q}$ many queries; for all $y \in \mathcal{D}_f$, it holds that $|\tilde{f}(y)| \leq \tilde{q}$. Using an averaging argument, we get the following lemma.

Lemma 15. *For all transcript τ, it holds that the number of popular points is at most $\tilde{q}2^{\frac{3n}{4}}$.*

Definition 14 (Good Point). *A point x is (r, ϵ_1)-good with respect to τ if it is (r, ϵ_1) robust and not popular with respect to τ.*

The following lemma is a corollary of Lemma 15 and the definition of the ϵ-crooked implementor. It says a random point is good with a high probability.

Lemma 16. *Let τ be a transcript. It holds that*

$$\Pr_{x \leftarrow \$ \mathcal{D}_f} [x \text{ is not } (r, \epsilon_1) \text{ good with respect to } \tau] \leq \epsilon_1^{\frac{1}{2}} + \frac{|\tau|}{2^n} + \frac{\tilde{q}}{2^{\frac{n}{4}}}.$$

Proof. Define $y_\zeta = f(x) + \zeta 0^{n-r}$ for $\zeta \in \{0,1\}^r$. From the definition of ϵ crooked implementor,

$$\Pr_{x \leftarrow \$ \mathcal{D}_f, f \leftarrow \mathsf{F}_{n,n|\tau}} \left[\bigvee_{\zeta \in \{0,1\}^r} y_\zeta \in C_{f,\tau} \right] \leq 2^r \epsilon_1.$$

By an averaging argument,

$$\Pr_{x \leftarrow \$ \mathcal{D}_f} \left[\Pr_{f \leftarrow \mathsf{F}_{n,n|\tau}} \left[\bigvee_{\zeta \in \{0,1\}^r} y_\zeta \in C_{f,\tau} \right] > 2^r \epsilon_1^{\frac{1}{2}} \right] \leq \epsilon_1^{\frac{1}{2}}.$$

We derive,

$$\begin{aligned} & \Pr_{x \leftarrow \$ \mathcal{D}_f} [x \text{ is not } (r, \epsilon_1) \text{ good with respect to } \tau] \\ = & \Pr_{x \leftarrow \$ \mathcal{D}_f} [x \text{ is not } (r, \epsilon_1) \text{ robust with respect to } \tau] \\ + & \Pr_{x \leftarrow \$ \mathcal{D}_f} [x \text{ is popular with respect to } \tau] \\ \leq & \epsilon_1^{\frac{1}{2}} + \frac{|\tau|}{2^n} + \frac{\tilde{q}}{2^{\frac{n}{4}}}. \end{aligned}$$

□

Next, we wish to ensure that all possible chaining values generated from a good point also become good points. We need the following definition.

Definition 15. *Let x be an (r, ϵ_1)-good point with respect to τ. We say y is eligible for (τ, x) if*

1. *y is an (r, ϵ_1)-good point with respect to τ.*
2. *for $\tau' = \tau \cup (x, y)$, it holds that y is (r, ϵ_1)-good point with respect to τ'.*

Now, we are ready to state our main tool.

Proposition 1. *Let x be ϵ_1-good point with respect to τ.*

$$\Pr_{y \leftarrow \$ \mathcal{D}_f} [y \text{ is not eligible with respect to } (\tau, x)] \leq \epsilon_1^{\frac{1}{2}} + \frac{|\tau|}{2^n} + \frac{\tilde{q} + 2^r}{2^{\frac{n}{4}}}.$$

Proof. The idea of the proof is to show that if we sample a point uniformly at random from $\mathcal{D}_f$, then by Lemma 16, with high probability, the point is (r, ϵ_1)-good with respect to τ. That means

$$\Pr_{f \leftarrow\$ \mathsf{F}_{n,n|\tau}}\left[\bigvee_{b' \in \{0,1\}^r} (f(y)b'0^{n-r} \in C_{f,\tau'})\right] \leq 2^r \left(\epsilon_1^{\frac{1}{2}} + \frac{|\tau|}{2^n} + \frac{|\tau|}{2^{\frac{n}{2}}}\right).$$

Now, if it also holds that $(f(y)b'0^{n-r}) \not\twoheadrightarrow x$ for any $b' \in \{0,1\}^r$, the point y will remain ϵ_1-good with respect to $\tau \cup (x, y)$. To prove it formally, we consider the following events.

1. **y-is-bad**: y is not (r, ϵ_1)-good with respect to τ.
2. **x-is-queried**: $\Pr_{f \leftarrow\$ \mathsf{F}_{n,n|\tau}}[(f(y)b'0^{n-r}) \twoheadrightarrow x] \geq \frac{1}{2^{\frac{n}{2}}}$ for some $b' \in \{0,1\}^r$.

The following lemma (to be proved in Section 6.5.2) says that if the above two events do not occur, then $f(y)$ is an (r, ϵ_1)-good point with respect to τ'.

Lemma 17. *Suppose y is such that the event* $\neg$**y-is-bad** $\bigwedge \neg$**x-is-queried** *holds. Then it holds that y is (r, ϵ_1)-good with respect to $\tau' = \tau \cup (x, y)$.*

$$\Pr_{f \leftarrow\$ \Gamma_{\tau'}}\left[\bigvee_{b' \in \{0,1\}^r} (f(y)b'0^{n-r} \in C_{f,\tau'})\right] \leq 2^r \left(\epsilon_1^{\frac{1}{2}} + \frac{|\tau| + 1}{2^n} + \frac{|\tau| + 1}{2^{\frac{n}{2}}}\right).$$

Thus to prove Proposition 1, we need to bound the probability of the events **y-is-bad** and **x-is-queried**. By Lemma 16,

$$\Pr_{y \leftarrow\$ \mathcal{D}_f}[\textbf{y-is-bad}] \leq \epsilon_1^{\frac{1}{2}} + \frac{|\tau|}{2^n} + \frac{\tilde{q}}{2^{\frac{n}{4}}}.$$

Finally, by the definition of popular points,

$$\Pr_{y \leftarrow\$ \mathcal{D}_f}[\textbf{x-is-queried}] = 2^r \Pr_{z \leftarrow\$ \mathcal{D}_f}[x \text{ is favourite of } z] \leq \frac{2^r}{2^{\frac{n}{4}}}.$$

This finishes the proof of Proposition 1. □

Bounding Probability of Winning Force-Crook. We are ready to bound the success probability of any adversary in the Force-Crook game against the Sponge construction when the underlying primitive is a random function $f : \{0,1\}^n \to \{0,1\}^n$. Specifically, we shall show that the adversary can not force a crooked chaining input for any query made by C.

Bad events. Recall that the adversary makes at most q_2 many queries to the oracle C^f. Each such query leads to $\ell + s$ many calls (referred to as chaining queries) to f made by C. We consider these chaining queries to be a sequence of $\sigma = q_2(\ell + s)$ many queries. By saying i-th query, we denote the i-th chaining query from this sequence. We consider the following bad events. The first bad event (**CrookedFirstInput**) occurs if, for any message, the *first* chaining value is crooked. We set the second bad event (**BadChain**) if, for some message queried by the distinguisher, we get a chaining value that is not (r, ϵ_1)-good as defined in Definition 14.

1. **CrookedFirstInput**. We say a bad event occurs if for the initial random R, for some $m_0 \in \{0,1\}^r$,

$$\Pr_{f \leftarrow\$ \Gamma_{\tilde{\tau}}}[Rm_0 0^{n-r} \in C_{f,\tilde{\tau}}] \geq 2^r \left(\epsilon_1^{\frac{1}{2}} + \frac{q_1}{2^n}\right).$$

2. **BadChain**. We say the i-th chaining query x_i raises bad event (denoted by $\mathbf{BadChain}_i$) if x_i is not (r, ϵ_1)-good with respect to the (up to that query) transcript τ. We define $\mathbf{BadChain} \stackrel{\text{def}}{=} \cup_{i=1}^{\sigma} \mathbf{BadChain}_i$.

Note that, for standard indifferentiability proofs, we usually consider a bad event when a chaining query input collides with some unchained query (made by the adversary to the oracle f) input. In our setting, such unchained queries are part of the transcript, and the definition of good points ensures that the chaining query does not result in such a collision.

Bounding Probabilities of Bad Events. First, we bound the probability of **CrookedFirstInput**. From the definition of ϵ-subversion, when the probabilities are taken over $f \leftarrow\!\$\ \Gamma_{\tilde{\tau}}$ and $x \leftarrow\!\$\ \mathcal{D}_f$

$$\Pr\left[x \in C_{f,\tilde{\tau}}\right] \leq \epsilon_1.$$

By an averaging argument, we get that

$$\Pr_{R \leftarrow \$ \mathcal{D}_f}\left[\Pr_{f \leftarrow \$ \Gamma_{\tilde{\tau}}}\left[R \in C_{f,\tilde{\tau}}\right] > \epsilon_1^{\frac{1}{2}}\right] \leq \epsilon_1^{\frac{1}{2}}.$$

Thus we bound

$$\Pr_{R \leftarrow \$ \mathcal{D}_f}[\mathbf{CrookedFirstInput}] \leq 2^r \epsilon_1^{\frac{1}{2}}. \tag{6.2}$$

Next, we bound $\Pr[\mathbf{BadChain}]$. For this case, we derive

$$\Pr[\mathbf{BadChain}] = \Pr[\mathbf{BadChain}_1] + \sum_{j=2}^{\sigma} \Pr[\mathbf{BadChain}_j \mid \bigwedge_{j'=1}^{j-1} \neg\mathbf{BadChain}_{j'}].$$

We start with bounding $\Pr[\mathbf{BadChain}_1]$. As R is uniformly chosen, from Lemma 16

$$\begin{aligned}\Pr[\mathbf{BadChain}_1] &= \Pr_{R \leftarrow \$ \mathcal{D}_f}\left[R \text{ is not } (r, \epsilon_1) \text{ -good with respect to. } \tilde{\tau}\right] \\ &\leq \epsilon_1^{\frac{1}{2}} + \frac{q_1}{2^n} + \frac{\tilde{q}}{2^{\frac{n}{4}}}.\end{aligned}$$

Consider the i-th chaining query x_i where $i > 1$. Let τ_i denote the transcript up to i-th query. We find the chaining query x_k, queried before x_i $(k < i)$ such that

$$x_i = f(x_k)b0^{n-r} \qquad \text{for some } b \in \{0,1\}^r.$$

Given $\bigwedge_{j'=1}^{i-1} \neg\mathbf{BadChain}_{j'}$, we conclude x_k is (r, ϵ_1)-good. If $f(x_k)b0^{n-r}$ is not (r, ϵ_1)-good with respect to τ_{k+1}, this means $f(x_k)b0^{n-r}$ was not eligible with respect to (τ_k, x_k) for some $b \in \{0,1\}^r$. Using Proposition 1,

$$\begin{aligned}&\Pr_{f \leftarrow \$ \Gamma_{\tau_k}}\left[\bigvee_{b \in \{0,1\}^r} (f(x_k)b0^{n-r}) \text{ is not eligible w.r.t. } (\tau_k, x_k)\right] \\ &\leq 2^r\left(\epsilon_1^{\frac{1}{2}} + \frac{\tilde{q} + 2^r}{2^{\frac{n}{4}}} + \frac{k}{2^n}\right).\end{aligned}$$

Thus we get

$$\Pr[\mathbf{BadChain}_j \mid \bigwedge_{j'=1}^{j-1} \neg\mathbf{BadChain}_{j'}] \leq 2^r\left(\epsilon_1^{\frac{1}{2}} + \frac{\tilde{q} + 2^r}{2^{\frac{n}{4}}} + \frac{j}{2^n}\right).$$

Taking the sum over all j we get

$$\Pr[\mathbf{BadChain}] \leq 2^r \left(\sigma \epsilon_1^{\frac{1}{2}} + \frac{\sigma(\tilde{q} + 2^r)}{2^{\frac{n}{4}}} + \frac{\sigma^2}{2^n} \right). \tag{6.3}$$

Bounding the $\mathsf{Force\text{-}Crook}$ Advantage. Let W_i denote the event that the input to the i-th query is crooked.

$$\begin{aligned} &\Pr[\mathcal{A} \text{ wins the game } \mathsf{Force\text{-}Crook}] \\ &\quad \leq \Pr[\mathbf{CrookedFirstInput}] + \Pr[\mathbf{BadChain}] \\ &\quad + \sum_{i=1}^{\sigma} \Pr[W_i \mid \neg\mathbf{CrookedFirstInput} \bigwedge \neg\mathbf{BadChain}]. \end{aligned}$$

As we already have the bound on the probabilities of the bad events, we need to bound

$$\Pr\left[W_i \mid \neg\mathbf{CrookedFirstInput} \bigwedge \neg\mathbf{BadChain}\right].$$

Consider the i-th chaining query x_i where $i > 1$. We find the chaining query x_k previous to x_i $(k < i)$. As $\neg\mathbf{BadChain}$ holds, x_k is (r, ϵ')-good with respect to the partial transcript τ_k.

$$\Pr_{f \leftarrow\$ \Gamma_{\tau_k}} \left[\bigvee_{b \in \{0,1\}^r} (f(x_k) b 0^{n-r} \in C_{f,\tau_k}) \right] \leq 2^r \left(\epsilon_1^{\frac{1}{2}} + \frac{k}{2^n} + \frac{k}{2^{\frac{n}{2}}} \right).$$

This implies

$$\Pr_{f \leftarrow\$ \Gamma_{\tau_k}} [x_i \in C_{f,\tau_k}] \leq 2^r \left(\epsilon_1^{\frac{1}{2}} + \frac{k}{2^n} + \frac{k}{2^{\frac{n}{2}}} \right) \leq 2^r \left(\epsilon_1^{\frac{1}{2}} + \frac{i}{2^n} + \frac{i}{2^{\frac{n}{2}}} \right).$$

As the responses of all the f queries are answered truthfully, for a $f \leftarrow\$ \Gamma_{\tau_k}$, $f \cup \tau_k$ is a uniform random element of $\Gamma_{\tilde{\tau}}$. Thus we get

$$\Pr_{f \leftarrow\$ \Gamma_{\tilde{\tau}}} [x_i \in C_{f,\tilde{\tau}} \mid \neg\mathbf{CrookedFirstInput} \wedge \neg\mathbf{BadChain}] \leq 2^r \left(\epsilon_1^{\frac{1}{2}} + \frac{i}{2^n} + \frac{i}{2^{\frac{n}{2}}} \right).$$

Taking the sum over all i, we get

$$\begin{aligned} \sum_{i=1}^{\sigma} \Pr_{f \leftarrow\$ \Gamma_{\tilde{\tau}}} &[W_i \mid \neg\mathbf{CrookedFirstInput} \wedge \neg\mathbf{BadChain}] \\ &\leq \sum_{i=1}^{\sigma} 2^r \left(\epsilon_1^{\frac{1}{2}} + \frac{i}{2^n} + \frac{i}{2^{\frac{n}{2}}} \right) \\ &\leq 2^r \left(\sigma \epsilon_1^{\frac{1}{2}} + \frac{\sigma^2}{2^n} + \frac{\sigma^2}{2^{\frac{n}{2}}} \right). \end{aligned} \tag{6.4}$$

Finally, adding Inequalities (6.2),(6.3), and (6.4) we get

$$\begin{aligned} &\Pr[\mathcal{A} \text{ wins the game } \mathsf{Force\text{-}Crook}] \\ &\leq \mathcal{O}\left(2^r \times \sigma \times \left(\epsilon_1^{\frac{1}{2}} + \frac{(\tilde{q} + 2^r)}{2^{\frac{n}{4}}} + \epsilon_1^{\frac{1}{2}} + \frac{\sigma}{2^n} + \frac{\sigma}{2^{\frac{n}{2}}} \right) \right). \end{aligned}$$

This finishes the proof of Lemma 14 and thus the proof of Theorem 5.

6.5.2 Proof of Lemma 17

Proof. Lemma 17 considers a transcript τ and points $x, y \in \{0,1\}^n$. Suppose y is such that the condition ($\neg$**y-is-bad** $\bigwedge \neg$**x-is-queried**) holds. The condition ($\neg$**y-is-bad**) implies that y is a (r, ϵ_1)-good point with respect to τ. The lemma says that y is a (r, ϵ_1)-good point with respect to $\tau' = \tau \cup (x, y)$.
Given the conditions and following Definition 14, we get that y is (r, ϵ_1)-robust with respect to τ and y is not popular. By Definition 12 we have

$$\Pr_{f \leftarrow\$ \mathsf{F}_{n,n|\tau}}\left[\bigvee_{b' \in \{0,1\}^r} (f(y)b'0^{n-r} \in C_{f,\tau})\right] \leq 2^r \left(\epsilon_1^{\frac{1}{2}} + \frac{|\tau|}{2^n} + \frac{|\tau|}{2^{\frac{n}{2}}}\right).$$

Our target is to bound the probability that y is not (r, ϵ_1)-good with respect to τ'. Let $Y_{b'}$ denote $f(y)b'0^{n-r}$. First, we bound the probability (over $f \leftarrow\$ \Gamma_{\tau'}$) that $Y_{b'}$ is not a (r, ϵ_1)-robust point with respect to $\tau' = \tau \cup (x, y)$. We have two cases:

a) $Y_{b'} = x$ for some $b' \in \{0,1\}^r$,

b) $Y_{b'} \in C_{f,\tau}$ for some $b' \in \{0,1\}^r$.

By union bound

$$\Pr\left[\bigvee_{b' \in \{0,1\}^r} (Y_{b'} \in C_{f,\tau'})\right] \leq \Pr\left[\bigvee_{b' \in \{0,1\}^r} (Y_{b'} = x)] + \Pr[\bigvee_{b' \in \{0,1\}^r} (Y_{b'} \in C_{f,\tau})\right]. \tag{6.5}$$

The term $\Pr[\bigvee_{b' \in \{0,1\}^r} (Y_{b'} = x)]$ is bounded above by $\frac{2^r}{2^n}$. For the second term, we can bound the probability (over $f \leftarrow\$ \Gamma_{\tau'}$) as

$$\begin{aligned}\Pr\left[\bigvee_{b' \in \{0,1\}^r} (Y_{b'} \in C_{f,\tau})\right] \leq &\Pr\left[\bigvee_{b' \in \{0,1\}^r} (Y_{b'} \twoheadrightarrow_f x)\right] + \\ &\Pr\left[\left(\bigvee_{b' \in \{0,1\}^r} (Y_{b'} \in C_{f,\tau})\right) \bigwedge \bigwedge_{b' \in \{0,1\}^r} (Y_{b'} \not\twoheadrightarrow_f x)\right].\end{aligned} \tag{6.6}$$

Bounding $\Pr\left[\bigvee_{b' \in \{0,1\}^r} (Y_{b'} \twoheadrightarrow_f x)\right]$. We first show that the probability that $Y_{b'}$ queries x is the same for all the transcripts, irrespective of where the value of $f(x)$ is set. In other words, we shall establish that the probability that $Y_{b'}$ queries x is the same in both transcripts τ and τ'.

$$\begin{aligned}\Pr_{f \leftarrow\$ \mathsf{F}_{n,n|\tau}}\left[\bigwedge_{b' \in \{0,1\}^r} (Y_{b'} \not\twoheadrightarrow_f x)\right] &= \sum_z \Pr_{f \leftarrow\$ \mathsf{F}_{n,n|\tau}}\left[\bigwedge_{b' \in \{0,1\}^r} (Y_{b'} \not\twoheadrightarrow_f x) \bigwedge f(x) = z\right] \\ &= 2^n \Pr_{f \leftarrow\$ \mathsf{F}_{n,n|\tau}}\left[\bigwedge_{b' \in \{0,1\}^r} (Y_{b'} \not\twoheadrightarrow_f x) \bigwedge f(x) = y\right] \\ &= \Pr_{f \leftarrow\$ \mathsf{F}_{n,n|\tau}}\left[\left(\bigwedge_{b' \in \{0,1\}^r} (Y_{b'} \not\twoheadrightarrow_f x)\right) \mid f(x) = y\right]\end{aligned}$$

$$= \Pr_{f \leftarrow \$ \Gamma_{\tau'}} \left[\bigwedge_{b' \in \{0,1\}^r} (Y_{b'} \not\twoheadrightarrow_f x) \right].$$

Now, taking the complement

$$\begin{aligned} \Pr_{f \leftarrow \$ \Gamma_{\tau'}} \left[\bigvee_{b' \in \{0,1\}^r} (Y_{b'} \twoheadrightarrow_f x) \right] &= 1 - \Pr_{f \leftarrow \$ \Gamma_{\tau'}} \left[\bigwedge_{b' \in \{0,1\}^r} (Y_{b'} \twoheadrightarrow_f x) \right] \\ &= 1 - \Pr_{f \leftarrow \$ \mathsf{F}_{n,n|\tau}} \left[\bigwedge_{b' \in \{0,1\}^r} (Y_{b'} \twoheadrightarrow_f x) \right] \\ &= \Pr_{f \leftarrow \$ \mathsf{F}_{n,n|\tau}} \left[\bigvee_{b' \in \{0,1\}^r} (Y_{b'} \twoheadrightarrow_f x) \right]. \end{aligned}$$

Bounding $\Pr\left[\left(\bigvee_{b' \in \{0,1\}^r} (Y_{b'} \in C_{f,\tau})\right) \wedge \bigwedge_{b' \in \{0,1\}^r} (Y_{b'} \not\twoheadrightarrow_f x)\right]$. Similar to the first case, we show the probability is identical for both transcripts.

$$\begin{aligned} &\Pr_{f \leftarrow \$ \mathsf{F}_{n,n|\tau}} \left[\left(\bigvee_{b' \in \{0,1\}^r} (Y_{b'} \in C_{f,\tau}) \right) \bigwedge \bigwedge_{b' \in \{0,1\}^r} (Y_{b'} \not\twoheadrightarrow_f x) \right] \\ &= \sum_z \Pr_{f \leftarrow \$ \mathsf{F}_{n,n|\tau}} \left[\left(\bigvee_{b' \in \{0,1\}^r} (Y_{b'} \in C_{f,\tau}) \right) \bigwedge \bigwedge_{b' \in \{0,1\}^r} (Y_{b'} \not\twoheadrightarrow_f x) \bigwedge f(x) = z \right] \\ &= 2^n \Pr_{f \leftarrow \$ \mathsf{F}_{n,n|\tau}} \left[\left(\bigvee_{b' \in \{0,1\}^r} (Y_{b'} \in C_{f,\tau}) \right) \bigwedge \bigwedge_{b' \in \{0,1\}^r} (Y_{b'} \not\twoheadrightarrow_f x) \bigwedge f(x) = y \right] \\ &= \Pr_{f \leftarrow \$ \mathsf{F}_{n,n|\tau}} \left[\left(\bigvee_{b' \in \{0,1\}^r} (Y_{b'} \in C_{f,\tau}) \right) \bigwedge \bigwedge_{b' \in \{0,1\}^r} (Y_{b'} \not\twoheadrightarrow_f x) \mid f(x) = y \right] \\ &= \Pr_{f \leftarrow \$ \Gamma_{\tau'}} \left[\left(\bigvee_{b' \in \{0,1\}^r} (Y_{b'} \in C_{f,\tau}) \right) \bigwedge \bigwedge_{b' \in \{0,1\}^r} (Y_{b'} \not\twoheadrightarrow_f x) \right]. \end{aligned}$$

The Final Derivation. Now we are ready to bound $\Pr_{f \leftarrow \$ \Gamma_{\tau'}}[\bigvee_{b' \in \{0,1\}^r} (Y_{b'} \in C_{f,\tau'})]$. In the following derivation, we use inequality 6.5 in the first step, inequality 6.6 in the second step, and the above two cases in the third step.

$$\begin{aligned} &\Pr_{f \leftarrow \$ \Gamma_{\tau'}} \left[\bigvee_{b' \in \{0,1\}^r} (Y_{b'} \in C_{f,\tau'}) \right] \\ &\leq \Pr_{f \leftarrow \$ \Gamma_{\tau'}} \left[\bigvee_{b' \in \{0,1\}^r} (Y_{b'} = x) \right] + \Pr_{f \leftarrow \$ \Gamma_{\tau'}} \left[\bigvee_{b' \in \{0,1\}^r} (Y_{b'} \in C_{f,\tau}) \right] \\ &\leq \frac{2^r}{2^n} + \Pr_{f \leftarrow \$ \Gamma_{\tau'}} \left[\bigvee_{b' \in \{0,1\}^r} (Y_{b'} \twoheadrightarrow_f x) \right] \end{aligned}$$

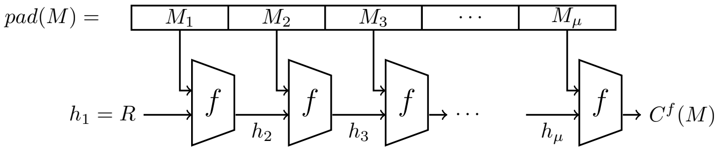


Figure 6.6: Merkle-Damgård mode of operation with random IV

$$
\begin{aligned}
&+ \Pr_{f \leftarrow\$ \Gamma_{\tau'}}\left[\left(\bigvee_{b' \in \{0,1\}^r} (Y_{b'} \in C_{f,\tau})\right) \bigwedge \bigwedge_{b' \in \{0,1\}^r} (Y_{b'} \not\twoheadrightarrow_f x)\right] \\
&= \frac{2^r}{2^n} + \Pr_{f \leftarrow\$ \mathsf{F}_{n,n|\tau}}\left[\bigvee_{b' \in \{0,1\}^r} (Y_{b'} \twoheadrightarrow_f x)\right] \\
&+ \Pr_{f \leftarrow\$ \mathsf{F}_{n,n|\tau}}\left[\left(\bigvee_{b' \in \{0,1\}^r} (Y_{b'} \in C_{f,\tau})\right) \bigwedge \bigwedge_{b' \in \{0,1\}^r} (Y_{b'} \not\twoheadrightarrow_f x)\right] \\
&\leq \frac{2^r}{2^n} + \Pr_{f \leftarrow\$ \mathsf{F}_{n,n|\tau}}\left[\bigvee_{b' \in \{0,1\}^r} (Y_{b'} \twoheadrightarrow_f x)\right] + \Pr_{f \leftarrow\$ \mathsf{F}_{n,n|\tau}}\left[\left(\bigvee_{b' \in \{0,1\}^r} (Y_{b'} \in C_{f,\tau})\right)\right] \\
&\leq \frac{2^r}{2^n} + \frac{2^r}{2^{\frac{n}{2}}} + 2^r\left(\epsilon_1^{\frac{1}{2}} + \frac{|\tau|}{2^n} + \frac{|\tau|}{2^{\frac{n}{2}}}\right).
\end{aligned}
$$

In the last line, we used, as the event (¬**y-is-bad**) holds,

$$
\Pr_{f \leftarrow\$ \mathsf{F}_{n,n|\tau}}\left[\left(\bigvee_{b' \in \{0,1\}^r} (Y_{b'} \in C_{f,\tau})\right)\right] \leq 2^r\left(\epsilon_1^{\frac{1}{2}} + \frac{|\tau|}{2^n} + \frac{|\tau|}{2^{\frac{n}{2}}}\right).
$$

and as the event (¬**x-is-queried**) holds

$$
\Pr_{f \leftarrow\$ \mathsf{F}_{n,n|\tau}}\left[\bigvee_{b' \in \{0,1\}^r} (Y_{b'} \twoheadrightarrow_f x)\right] \leq \frac{2^r}{2^{\frac{n}{2}}}.
$$

□

6.6 Crooked Indifferentiability of Merkle-Damgård

In this section, we show that the classical Merkle-Damgård construction using $n+1$-to-n-bit compression function f and instantiated with a random initialisation vector is crooked-indifferentiable from a random oracle.

Merkle-Damgård Construction. The details of the parameters of Merkle-Damgård construction are listed below. The construction is shown in Figure 6.6

TARGET HASH FUNCTION. The construction implements a hash function $H : \{0,1\}^\mu \to \{0,1\}^n$.

PRIMITIVES. The underlying primitive of the construction is an $n+1$-to-n bit function $f : \{0,1\}^{n+1} \to \{0,1\}^n$.

PUBLIC RANDOMNESS. The public randomness is $R \leftarrow\$ \{0,1\}^n$.

MESSAGE PREPROCESSING. The indifferentiability of Merkle-Damgård requires the message space to be prefix-free. We assume the same. Note if we consider the fixed input length hash function, we do not need any prefix-free padding. The input message $M \in \{0,1\}^\mu$ is parsed as bits $M_1 M_2 \ldots M_\mu$.
Our main result in this section is Theorem 7.

Theorem 7. *Let $f : \{0,1\}^{n+1} \to \{0,1\}n$ be a random function and $C^f : \{0,1\}^\mu \to \{0,1\}^n$ be the Merkle-Damgård construction. There exists a simulator S such that for all $((q_1, \tilde{q}, \epsilon), q_2)$-crooked distinguisher $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$*

$$\mathbf{Adv}^{\mathsf{crooked\text{-}indiff}}_{\mathcal{A},(C,f)} \leq \mathcal{O}\left(\sigma \times \left(\epsilon_1^{\frac{1}{2}} + \frac{\tilde{q}}{2^{\frac{n}{4}}} + \epsilon_1^{\frac{1}{2}} + \frac{\sigma}{2^n} + \frac{\sigma}{2^{\frac{n}{2}}}\right)\right)$$

where $\epsilon_1 = \epsilon + \frac{q_1}{2^n}$ q_2 is the total number of construction queries made by $\mathcal{A}_2$ and σ is the total number of blocks in the queries made by $\mathcal{A}_2$.

The Theorem follows from Theorem 8 and Lemma 18.
Classical Indifferentiability of Merkle-Damgård Construction. We recall the classical indifferentiability result of Merkle-Damgård mode from [36] in our notations.

Theorem 8 (Theorem 3.1 in [36])**.** *Prefix-free Merkle-Damgård mode instantiating $C^f : \{0,1\}^\mu \to \{0,1\}^n$ is (q_2, q_{sim}, δ_i)-indifferentiable from a random oracle for $q_{sim} = \mathcal{O}(\sigma^2)$ and $\delta_i = \mathcal{O}(\frac{\sigma^2}{2^n})$ where σ is the total number of blocks in the queries made by the distinguisher.*

Bounding Probability of Winning Force-Crook.

Lemma 18. *Let C be the Merkle-Damgård construction considered in this section.*

$$\mathbf{Insec}^{\mathsf{force-crook}}_{C,(q_1,\tilde{q}),(q_2,q_s)} \leq \mathcal{O}\left(\sigma \times \left(\epsilon_1^{\frac{1}{2}} + \frac{\tilde{q}}{2^{\frac{n}{4}}} + \epsilon_1^{\frac{1}{2}} + \frac{\sigma}{2^n} + \frac{\sigma}{2^{\frac{n}{2}}}\right)\right)$$

where $\sigma = q_2\mu + q_S$.

The proof of the lemma works exactly as the proof of Lemma 14. The only difference is in the parameters of the definitions. We skip the proof.

6.7 Concluding Discussion

We wish to finish the chapter with some discussion on the possibility and challenges of extending our proof to Sponge construction with permutations. Finally, we present some research directions we find interesting.

6.7.1 Sponge Construction Based on Permutation

The reader may note that the Sponge construction in practice is based on a fixed *permutation* where the adversary is allowed to make inverse queries. We attempted to extend our proof for the permutations as well but could not solve one key issue. One main step (Proposition 1) in our proof was to show that a good point y with respect to a partial transcript τ remains a good point if another good point x is mapped to y. To prove that, we argued that the queries of $\tilde{f}(y)$ and $\tilde{f}(f(y))$ are independent from the preimage of y. Thus, we could include a good point and extend the transcript without invoking bad.

This argument does not hold when f is a permutation. In that case $\tilde{f}$ can indeed make f^{-1} queries. Extending the transcript with good points and simultaneously handling inverse queries seem to require a different technique. One could try adding additional ingredients like xoring independent random strings in each iteration. However, that would increase the number of random strings to be linear with the message length, and the resulting construction would not be practical.

Chapter 7

Concluding Discussion and Future Research Directions.

In this thesis, we have discussed indifferentiability and some of the other related security notions in detail. We have investigated constructions which achieve some pre-stated security goals, and are of practical importance. We can summarise our contributions and some future research directions which can be investigated building upon our work as follows:

- We have shown that the TLR3 construction introduced by Coron et al. in [37] is secure in the indifferentiability model up to almost 2^n queries; in establishing this improved bound we have taken forward the work that begun in [37] and [65]. A future direction of research can be to investigate whether our bounds are tight, i.e., if there exists an adversary which can attack TLR3 in the indifferentiability model, making $\mathcal{O}(2^n)$ queries.

- Next, we study the security analysis of a five-round ideal KAF cipher based on five independent public round permutations and five independent round keys. We show that the construction is secure up to almost $2^{\frac{2n}{3}}$ queries. However, we believe that one can reduce the number of keys and round permutations of the construction and achieve a similar security bound. Unfortunately, the security proof for such a construction will be extremely tedious due to the increased degree of input-output dependency at each round, which forces one to use technical machinery like the sum-capture lemma [31] and its variants [95] in the security proof. Establishing the tightness of the proven bound or improving the bound of the construction from $2n/3$-bits to $3n/4$-bits is also left as a future research problem.

- Next, we investigate the crooked indifferentiability security notion in detail. Subversion Resistance of the hash function is an important security property when used to replace random oracles in the kleptographic setting. This work is the first to analyse the security of practically used hashing modes in the crooked indifferentiability framework. Our techniques show how to prove crooked indifferentiability when the underlying primitive is modelled as a random function. At first, We discuss the enveloped xor construction as defined in [90] and discover some errors in their proof. We then present a corrected proof for the crooked indifferentiability of the envelope xor construction. We then try to simplify the process to prove the crooked indifferentiability of some construction and introduce the force-crook game, which bridges the gap between classical indifferentiability and crooked indifferentiability. We use the force-crook game to show that a version of the Sponge construction and a version of the Merkle-Damgård construction achieve crooked indifferentiability from a random oracle both of which are relatively cheaper and easier to implement than the enveloped xor construction. The first natural research problem would be to consider the crooked indifferentiability of Sponge construction in the random permutation model. It would also be interesting to consider proving the crooked indifferentiability of the ideal cipher constructions like the Feistel Network.

7.1 Concluding Remark

Indifferentiability is a greatly useful tool, which can be used to investigate the security of many real world cryptographic schemes, especially if we assume that the primitives used in building the scheme are publicly accessible. That doesn't mean that indifferentiability provides all the answers we look for in such scenarios. As Ristenpart et al. in [86] showed, the composition theorem for indifferentiability holds only for single-stage games and not for multi-stage games. Towards this end, they proposed the security notion of Reset indifferentiability. While the stronger version of this security notion, where one needs one simulator to work for all distinguishers is subject to significant impossibility results [5], the weaker version, where one uses different simulators for different distinguishers is still thought to be useful [103]. Zhandry recently showed that quantum indifferentiability is achievable [102]. New techniques have come up to prove the quantum indifferentiability of schemes, but still, the task remains challenging, particularly in the case where inverse queries are allowed. As we all know quantum computers are a very real threat in today's world and thus coming up with new techniques to prove quantum indifferentiability and designing quantum indiffrentiable schemes can be a worthwhile and rewarding future goal.

Bibliography

[1] Sha-1. [Online; accessed 21-December-2023].

[2] G. Ateniese, D. Francati, B. Magri, and D. Venturi. Public immunization against complete subversion without random oracles. In R. H. Deng, V. Gauthier-Umaña, M. Ochoa, and M. Yung, editors, *Applied Cryptography and Network Security - 17th International Conference, ACNS 2019, Bogota, Colombia, June 5-7, 2019, Proceedings*, volume 11464 of *Lecture Notes in Computer Science*, pages 465–485. Springer, 2019.

[3] G. Ateniese, A. Kiayias, B. Magri, Y. Tselekounis, and D. Venturi. Secure outsourcing of cryptographic circuits manufacturing. In J. Baek, W. Susilo, and J. Kim, editors, *Provable Security*, pages 75–93, Cham, 2018. Springer International Publishing.

[4] G. Ateniese, B. Magri, and D. Venturi. Subversion-resilient signatures: Definitions, constructions and applications. *Theor. Comput. Sci.*, 820:91–122, 2020.

[5] P. Baecher, C. Brzuska, and A. Mittelbach. Reset indifferentiability and its consequences. Cryptology ePrint Archive, Paper 2013/459, 2013. https://eprint.iacr.org/2013/459.

[6] M. Barbosa and P. Farshim. The related-key analysis of feistel constructions. In C. Cid and C. Rechberger, editors, *FSE 2014. Revised Selected Papers*, volume 8540 of *LNCS*, pages 265–284. Springer, 2014.

[7] R. Beaulieu, S. Treatman-Clark, D. Shors, B. Weeks, J. Smith, and L. Wingers. The simon and speck lightweight block ciphers. In *2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, pages 1–6, 2015.

[8] M. Bellare and V. T. Hoang. Resisting randomness subversion: Fast deterministic and hedged public-key encryption in the standard model. In E. Oswald and M. Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015*, pages 627–656, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.

[9] M. Bellare, K. G. Paterson, and P. Rogaway. Security of symmetric encryption against mass surveillance. In J. A. Garay and R. Gennaro, editors, *Advances in Cryptology – CRYPTO 2014*, pages 1–19, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[10] M. Bellare and P. Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In S. Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006*, pages 409–426, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[11] D. J. Bernstein, S. Kölbl, S. Lucks, P. M. C. Massolino, F. Mendel, K. Nawaz, T. Schneider, P. Schwabe, F. Standaert, Y. Todo, and B. Viguier. Gimli : A cross-platform permutation. In *CHES 2017, Proceedings*, pages 299–320, 2017.

[12] G. Bertoni, J. Daemen, M. Peeters, and G. V. Assche. Keccak. In *EUROCRYPT 2013. Proceedings*, pages 313–314, 2013.

[13] G. Bertoni, J. Daemen, M. Peeters, and G. Van Assche. On the indifferentiability of the sponge construction. In N. Smart, editor, *Advances in Cryptology – EUROCRYPT 2008*, 2008.

[14] A. Bhattacharjee, R. Bhaumik, A. Dutta, M. Nandi, and A. Raychaudhuri. Bbb security for 5-round even-mansour-based key-alternating feistel ciphers. *Des. Codes Cryptography*, 92(1):13–49, oct 2023.

[15] A. Bhattacharjee, C. M. López, E. List, and M. Nandi. The oribatida v1.3 family of lightweight authenticated encryption schemes. *J. Math. Cryptol.*, 15(1):305–344, 2021.

[16] S. Bhattacharya and M. Nandi. *Full Indifferentiable Security of the Xor of Two or More Random Permutations Using the χ^2 Method*, pages 387–412. 01 2018.

[17] R. Bhattacharyya and A. Mandal. On the indifferentiability of fugue and luffa. In J. Lopez and G. Tsudik, editors, *Applied Cryptography and Network Security*, pages 479–497, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[18] R. Bhattacharyya, A. Mandal, and M. Nandi. Indifferentiability characterization of hash functions and optimal bounds of popular domain extensions. In B. Roy and N. Sendrier, editors, *Progress in Cryptology - INDOCRYPT 2009*, pages 199–218, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[19] R. Bhattacharyya, A. Mandal, and M. Nandi. Security analysis of the mode of jh hash function. In S. Hong and T. Iwata, editors, *Fast Software Encryption*, pages 168–191, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[20] R. Bhattacharyya, M. Nandi, and A. Raychaudhuri. Crooked indifferentiability of enveloped xor revisited. In A. Adhikari, R. Küsters, and B. Preneel, editors, *Progress in Cryptology – INDOCRYPT 2021*, pages 73–92, Cham, 2021. Springer International Publishing.

[21] R. Bhattacharyya, M. Nandi, and A. Raychaudhuri. Subversion resilient hashing: Efficient constructions and modular proofs for crooked indifferentiability. *IEEE Trans. Inf. Theory*, 69(5):3302–3315, 2023.

[22] R. Bhaumik, M. Nandi, and A. Raychaudhuri. Improved indifferentiability security proof for 3-round tweakable luby–rackoff. *Des. Codes Cryptography*, 89(10):2255–2281, oct 2021.

[23] E. Biham and A. Shamir. Differential cryptanalysis of des-like cryptosystems. In A. Menezes and S. A. Vanstone, editors, *CRYPTO '90, Proceedings*, volume 537 of *LNCS*, pages 2–21. Springer, 1990.

[24] A. Bogdanov, M. Knezevic, G. Leander, D. Toz, K. Varici, and I. Verbauwhede. SPONGENT: the design space of lightweight cryptographic hashing. *IEEE Trans. Computers*, 62(10):2041–2053, 2013.

[25] A. Chakraborti, N. Datta, M. Nandi, and K. Yasuda. Beetle family of lightweight and secure authenticated encryption ciphers. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(2):218–241, 2018.

[26] B. Chakraborty and M. Nandi. Orange. *NIST LWC*, 2019.

[27] D. Chakraborty and P. Sarkar. A new mode of encryption providing a tweakable strong pseudo-random permutation. In M. Robshaw, editor, *Fast Software Encryption*, pages 293–309, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[28] D. Chang and M. Nandi. Improved indifferentiability security analysis of chopmd hash function. In K. Nyberg, editor, *Fast Software Encryption*, pages 429–443, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[29] S. Checkoway, R. Niederhagen, A. Everspaugh, M. Green, T. Lange, T. Ristenpart, D. J. Bernstein, J. Maskiewicz, H. Shacham, and M. Fredrikson. On the practical exploitability of dual EC in TLS implementations. In *23rd USENIX Security Symposium (USENIX Security 14)*, pages 319–335, San Diego, CA, Aug. 2014. USENIX Association.

[30] S. Chen, R. Lampe, J. Lee, Y. Seurin, and J. P. Steinberger. Minimizing the two-round even-mansour cipher. In J. A. Garay and R. Gennaro, editors, *CRYPTO 2014, Proceedings, Part I*, volume 8616 of *LNCS*, pages 39–56. Springer, 2014.

[31] S. Chen and J. P. Steinberger. Tight security bounds for key-alternating ciphers. In P. Q. Nguyen and E. Oswald, editors, *EUROCRYPT 2014. Proceedings*, volume 8441 of *LNCS*, pages 327–350. Springer, 2014.

[32] S. S. M. Chow, A. Russell, Q. Tang, M. Yung, Y. Zhao, and H.-S. Zhou. Let a non-barking watchdog bite: Cliptographic signatures with an offline watchdog. In D. Lin and K. Sako, editors, *Public-Key Cryptography – PKC 2019*, pages 221–251, Cham, 2019. Springer International Publishing.

[33] B. Cogliati, R. Lampe, and Y. Seurin. Tweaking even-mansour ciphers. In R. Gennaro and M. Robshaw, editors, *CRYPTO 2015, Proceedings, Part I*, volume 9215 of *LNCS*, pages 189–208. Springer, 2015.

[34] B. Cogliati and Y. Seurin. Beyond-birthday-bound security for tweakable even-mansour ciphers with linear tweak and key mixing. In T. Iwata and J. H. Cheon, editors, *ASIACRYPT 2015, Proceedings, Part II*, volume 9453 of *LNCS*, pages 134–158. Springer, 2015.

[35] S. Coretti, Y. Dodis, S. Guo, and J. P. Steinberger. Random oracles and non-uniformity. In J. B. Nielsen and V. Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part I*, volume 10820 of *Lecture Notes in Computer Science*, pages 227–258. Springer, 2018.

[36] J.-S. Coron, Y. Dodis, C. Malinaud, and P. Puniya. Merkle-damgård revisited: How to construct a hash function. In V. Shoup, editor, *Advances in Cryptology – CRYPTO 2005*, 2005.

[37] J.-S. Coron, Y. Dodis, A. Mandal, and Y. Seurin. A domain extender for the ideal cipher. In D. Micciancio, editor, *Theory of Cryptography*, pages 273–289, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[38] J.-S. Coron, T. Holenstein, R. Künzler, J. Patarin, Y. Seurin, and S. Tessaro. How to build an ideal cipher: The indifferentiability of the feistel construction. *Journal of Cryptology*, 29(1):61–114, Jan 2016.

[39] J.-S. Coron, J. Patarin, and Y. Seurin. The random oracle model and the ideal cipher model are equivalent. In D. Wagner, editor, *Advances in Cryptology – CRYPTO 2008*, pages 1–20, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[40] J. Daemen, S. Hoffert, M. Peeters, G. V. Assche, and R. V. Keer. Xoodyak, a lightweight cryptographic scheme. *IACR Trans. Symmetric Cryptol.*, 2020(S1):60–87, 2020.

[41] J. Daemen and V. Rijmen. The rijndael block cipher. [Online; accessed 21-December-2023].

[42] Y. Dai, Y. Seurin, J. Steinberger, and A. Thiruvengadam. Indifferentiability of iterated even-mansour ciphers with non-idealized key-schedules: Five rounds are necessary and sufficient. In J. Katz and H. Shacham, editors, *Advances in Cryptology – CRYPTO 2017*, pages 524–555, Cham, 2017. Springer International Publishing.

[43] Y. Dai and J. Steinberger. Indifferentiability of 8-round feistel networks. In M. Robshaw and J. Katz, editors, *Advances in Cryptology – CRYPTO 2016*, pages 95–120, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.

[44] J. P. Degabriele, P. Farshim, and B. Poettering. A more cautious approach to security against mass surveillance. In G. Leander, editor, *Fast Software Encryption*, pages 579–598, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.

[45] J. P. Degabriele, K. G. Paterson, J. C. N. Schuldt, and J. Woodage. Backdoors in pseudorandom number generators: Possibility and impossibility results. In M. Robshaw and J. Katz, editors, *Advances in Cryptology – CRYPTO 2016*, pages 403–432, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.

[46] C. Dobraunig, M. Eichlseder, S. Mangard, F. Mendel, B. Mennink, R. Primas, and T. Unterluggauer. Isap v2.0. *IACR Trans. Symmetric Cryptol.*, 2020(S1):390–416, 2020.

[47] C. Dobraunig, M. Eichlseder, F. Mendel, and M. Schläffer. Ascon v1.2. *NIST LWC*, 2019.

[48] Y. Dodis, C. Ganesh, A. Golovnev, A. Juels, and T. Ristenpart. A formal treatment of backdoored pseudorandom generators. In E. Oswald and M. Fischlin, editors, *Advances in Cryptology – EUROCRYPT 2015*, pages 101–126, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.

[49] Y. Dodis, S. Guo, and J. Katz. Fixing cracks in the concrete: Random oracles with auxiliary input, revisited. In J.-S. Coron and J. B. Nielsen, editors, *Advances in Cryptology – EUROCRYPT 2017*, pages 473–495, Cham, 2017. Springer International Publishing.

[50] Y. Dodis, L. Reyzin, R. L. Rivest, and E. Shen. Indifferentiability of permutation-based compression functions and tree-based modes of operation, with applications to md6. In O. Dunkelman, editor, *Fast Software Encryption*, pages 104–121, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[51] Y. Dodis, M. Stam, J. Steinberger, and T. Liu. Indifferentiability of confusion-diffusion networks. In M. Fischlin and J.-S. Coron, editors, *Advances in Cryptology – EUROCRYPT 2016*, pages 679–704, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.

[52] A. Dutta. Minimizing the two-round tweakable even-mansour cipher. In S. Moriai and H. Wang, editors, *ASIACRYPT 2020, Proceedings, Part I*, volume 12491 of *LNCS*, pages 601–629. Springer, 2020.

[53] S. Even and Y. Mansour. A construction of a cipher from a single pseudorandom permutation. *J. Cryptol.*, 10(3):151–162, 1997.

[54] C. Gentry and Z. Ramzan. Eliminating random permutation oracles in the even-mansour cipher. In P. J. Lee, editor, *ASIACRYPT 2004, Proceedings*, volume 3329 of *LNCS*, pages 32–47. Springer, 2004.

[55] C. Guo and L. Wang. Revisiting key-alternating feistel ciphers for shorter keys and multi-user security. In T. Peyrin and S. D. Galbraith, editors, *ASIACRYPT 2018, Proceedings, Part I*, volume 11272 of *LNCS*, pages 213–243. Springer, 2018.

[56] J. Guo, J. Jean, I. Nikolic, and Y. Sasaki. Meet-in-the-middle attacks on generic feistel constructions. In P. Sarkar and T. Iwata, editors, *ASIACRYPT 2014. Proceedings, Part I*, volume 8873 of *LNCS*, pages 458–477. Springer, 2014.

[57] J. Guo, T. Peyrin, and A. Poschmann. The PHOTON family of lightweight hash functions. In *CRYPTO 2011. Proceedings*, pages 222–239, 2011.

[58] V. T. Hoang and P. Rogaway. On generalized feistel networks. In T. Rabin, editor, *CRYPTO 2010. Proceedings*, volume 6223 of *LNCS*, pages 613–630. Springer, 2010.

[59] T. Holenstein, R. Künzler, and S. Tessaro. The equivalence of the random oracle model and the ideal cipher model, revisited. In *Proceedings of the Forty-third Annual ACM Symposium on Theory of Computing*, STOC '11, pages 89–98, New York, NY, USA, 2011. ACM.

[60] J. Jean. TikZ for Cryptographers. https://www.iacr.org/authors/tikz/, 2016.

[61] R. Lampe and Y. Seurin. Security analysis of key-alternating feistel ciphers. In C. Cid and C. Rechberger, editors, *FSE 2014. Revised Selected Papers*, volume 8540 of *LNCS*, pages 243–264. Springer, 2014.

[62] J. Lee. Indifferentiability of the sum of random permutations toward optimal security. *IEEE Transactions on Information Theory*, 63(6):4050–4054, June 2017.

[63] M. Luby and C. Rackoff. How to construct pseudorandom permutations from pseudorandom functions. *SIAM J. Comput.*, 17(2):373–386, 1988.

[64] A. Mandal, J. Patarin, and V. Nachef. Indifferentiability beyond the birthday bound for the xor of two public random permutations. In G. Gong and K. C. Gupta, editors, *Progress in Cryptology - INDOCRYPT 2010*, pages 69–81, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[65] A. Mandal, J. Patarin, and Y. Seurin. On the public indifferentiability and correlation intractability of the 6-round feistel construction. In *Proceedings of the 9th International Conference on Theory of Cryptography*, TCC'12, page 285–302, Berlin, Heidelberg, 2012. Springer-Verlag.

[66] M. Matsui. Linear cryptanalysis method for DES cipher. In T. Helleseth, editor, *EUROCRYPT '93, Proceedings*, volume 765 of *LNCS*, pages 386–397. Springer, 1993.

[67] U. Maurer, R. Renner, and C. Holenstein. Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology. In M. Naor, editor, *Theory of Cryptography Conference — TCC 2004*, volume 2951 of *Lecture Notes in Computer Science*, pages 21–39. Springer-Verlag, 2 2004.

[68] U. M. Maurer and K. Pietrzak. The security of many-round luby-rackoff pseudo-random permutations. In E. Biham, editor, *EUROCRYPT 2003, Proceedings*, volume 2656 of *LNCS*, pages 544–561. Springer, 2003.

[69] B. Mennink. Indifferentiability of double length compression functions. In M. Stam, editor, *Cryptography and Coding - 14th IMA International Conference, IMACC 2013, Oxford, UK, December 17-19, 2013. Proceedings*, volume 8308 of *Lecture Notes in Computer Science*, pages 232–251. Springer, 2013.

[70] I. Mironov and N. Stephens-Davidowitz. Cryptographic reverse firewalls. In E. Oswald and M. Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part II*, volume 9057 of *Lecture Notes in Computer Science*, pages 657–686. Springer, 2015.

[71] D. Moody, S. Paul, and D. Smith-Tone. Improved indifferentiability security bound for the JH mode. *Des. Codes Cryptography*, 79(2):237–259, May 2016.

[72] V. Nachef, J. Patarin, and E. Volte. *Feistel Ciphers - Security Proofs and Cryptanalysis.* Springer, 2017.

[73] Y. Naito. Indifferentiability of double-block-length hash function without feed-forward operations. In J. Pieprzyk and S. Suriadi, editors, *Information Security and Privacy - 22nd Australasian Conference, ACISP 2017, Auckland, New Zealand, July 3-5, 2017, Proceedings, Part II*, volume 10343 of *Lecture Notes in Computer Science*, pages 38–57. Springer, 2017.

[74] M. Nandi. The characterization of luby-rackoff and its optimum single-key variants. In G. Gong and K. C. Gupta, editors, *INDOCRYPT 2010. Proceedings*, volume 6498 of *LNCS*, pages 82–97. Springer, 2010.

[75] M. Nandi. On the optimality of non-linear computations of length-preserving encryption schemes. In T. Iwata and J. H. Cheon, editors, *ASIACRYPT 2015, Proceedings, Part II*, volume 9453 of *LNCS*, pages 113–133. Springer, 2015.

[76] M. Naor and O. Reingold. On the construction of pseudorandom permutations: Luby-rackoff revisited. *J. Cryptol.*, 12(1):29–66, 1999.

[77] J. Patarin. Pseudorandom permutations based on the DES scheme. In G. D. Cohen and P. Charpin, editors, *EUROCODE '90, Proceedings*, volume 514 of *LNCS*, pages 193–204. Springer, 1990.

[78] J. Patarin. How to construct pseudorandom and super pseudorandom permutations from one single pseudorandom function. In R. A. Rueppel, editor, *EUROCRYPT '92, Proceedings*, volume 658 of *LNCS*, pages 256–266. Springer, 1992.

[79] J. Patarin. About feistel schemes with six (or more) rounds. In S. Vaudenay, editor, *FSE '98, Proceedings*, volume 1372 of *LNCS*, pages 103–121. Springer, 1998.

[80] J. Patarin. Security of random feistel schemes with 5 or more rounds. In M. K. Franklin, editor, *CRYPTO 2004, Proceedings*, volume 3152 of *LNCS*, pages 106–122. Springer, 2004.

[81] J. Patarin. The “coefficients h” technique. In R. M. Avanzi, L. Keliher, and F. Sica, editors, *Selected Areas in Cryptography*, pages 328–345, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[82] J. Patarin. Security of balanced and unbalanced feistel schemes with linear non equalities. *IACR Cryptol. ePrint Arch.*, page 293, 2010.

[83] S. Patel, Z. Ramzan, and G. S. Sundaram. Towards making luby-rackoff ciphers optimal and practical. In L. R. Knudsen, editor, *FSE '99, Proceedings*, volume 1636 of *LNCS*, pages 171–185. Springer, 1999.

[84] B. Preneel. Cryptography in the post-snowden era. [Online; accessed 7-April-2021].

[85] Z. Ramzan and L. Reyzin. On the round security of symmetric-key cryptographic primitives. In M. Bellare, editor, *CRYPTO 2000, Proceedings*, volume 1880 of *LNCS*, pages 376–393. Springer, 2000.

[86] T. Ristenpart, H. Shacham, and T. Shrimpton. Careful with composition: Limitations of the indifferentiability framework. In K. G. Paterson, editor, *Advances in Cryptology – EUROCRYPT 2011*, pages 487–506, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[87] P. Rogaway, M. Bellare, and J. Black. Sha-3 standard. *(TISSEC)*, 6(3):365–403, 2003.

[88] A. Russell, Q. Tang, M. Yung, and H. Zhou. Cliptography: Clipping the power of kleptographic attacks. In J. H. Cheon and T. Takagi, editors, *Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part II*, volume 10032 of *Lecture Notes in Computer Science*, pages 34–64, 2016.

[89] A. Russell, Q. Tang, M. Yung, and H. Zhou. Generic semantic security against a kleptographic adversary. In B. Thuraisingham, D. Evans, T. Malkin, and D. Xu, editors, *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, pages 907–922. ACM, 2017.

[90] A. Russell, Q. Tang, M. Yung, and H. Zhou. Correcting subverted random oracles. In H. Shacham and A. Boldyreva, editors, *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part II*, volume 10992 of *Lecture Notes in Computer Science*, pages 241–271. Springer, 2018.

[91] A. Russell, Q. Tang, M. Yung, H. Zhou, and J. Zhu. Correcting subverted random oracles. *IACR Cryptol. ePrint Arch.*, page 42, 2021.

[92] B. Sadeghiyan and J. Pieprzyk. A construction for super pseudorandom permutations from A single pseudorandom function. In R. A. Rueppel, editor, *EUROCRYPT '92, Proceedings*, volume 658 of *LNCS*, pages 267–284. Springer, 1992.

[93] Y. Shen, H. Yan, L. Wang, and X. Lai. Secure key-alternating feistel ciphers without key schedule. *Sci. China Inf. Sci.*, 64(1), 2021.

[94] T. Suzaki, K. Minematsu, S. Morioka, and E. Kobayashi. Twine: A lightweight block cipher for multiple platforms. In L. R. Knudsen and H. Wu, editors, *SAC 2012, Revised Selected Papers*, volume 7707 of *LNCS*, pages 339–354. Springer, 2012.

[95] S. Tessaro and X. Zhang. Tight security for key-alternating ciphers with correlated subkeys. In M. Tibouchi and H. Wang, editors, *ASIACRYPT 2021, Proceedings, Part III*, volume 13092 of *LNCS*, pages 435–464. Springer, 2021.

[96] E. W. Weisstein. Bernoulli inequality. From MathWorld—A Wolfram Web Resource. [Online; accessed 30-November-2023].

[97] E. W. Weisstein. Markov's inequality. From MathWorld—A Wolfram Web Resource. [Online; accessed 7-April-2021].

[98] W. Wu and L. Zhang. Lblock: A lightweight block cipher. In J. López and G. Tsudik, editors, *ACNS 2011. Proceedings*, volume 6715 of *LNCS*, pages 327–344, 2011.

[99] Y. Wu, L. Yu, Z. Cao, and X. Dong. Tight security analysis of 3-round key-alternating cipher with a single permutation. In S. Moriai and H. Wang, editors, *ASIACRYPT 2020, Proceedings, Part I*, volume 12491 of *LNCS*, pages 662–693. Springer, 2020.

[100] A. L. Young and M. Yung. The dark side of "black-box" cryptography, or: Should we trust capstone? In N. Koblitz, editor, *Advances in Cryptology - CRYPTO '96, 16th Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 1996, Proceedings*, volume 1109 of *Lecture Notes in Computer Science*, pages 89–103. Springer, 1996.

[101] A. L. Young and M. Yung. Kleptography: Using cryptography against cryptography. In W. Fumy, editor, *Advances in Cryptology - EUROCRYPT '97, International Conference on the Theory and Application of Cryptographic Techniques, Konstanz, Germany, May 11-15, 1997, Proceeding*, volume 1233 of *Lecture Notes in Computer Science*, pages 62–74. Springer, 1997.

[102] M. Zhandry. How to record quantum queries, and applications to quantum indifferentiability. In A. Boldyreva and D. Micciancio, editors, *Advances in Cryptology – CRYPTO 2019*, pages 239–268, Cham, 2019. Springer International Publishing.

[103] M. Zhandry. Redeeming reset indifferentiability and post-quantum groups. Cryptology ePrint Archive, Paper 2021/288, 2021. https://eprint.iacr.org/2021/288.