

A Modern Day Approach to Combinatorial Secret Sharing

Anandarup Roy

A thesis presented for the degree of
Doctor of Philosophy in Computer Science
to the **Indian Statistical Institute**, Kolkata



Supervisor: Prof. Bimal Kumar Roy

Co-Supervisor: Prof. Mridul Nandi

Applied Statistics Unit
Indian Statistical Institute, Kolkata

July 2024

Acknowledgements

I admit the extraordinary debt I owe to all who supported me on my way to completing this thesis. The first person that I have to mention is my Ph.D. supervisor, Professor Bimal Kumar Roy for his invaluable guidance throughout my career and for his sustained patience and feedback during my darkest hours. His unwavering support in his busy schedule allowed me to explore further when there was no visible hope. Be it finding problems or collaborators, I enjoyed complete freedom. My co-supervisor Professor Mridul Nandi also requires a very special mention without whose help this thesis might not have materialised. To work under their guidance further would be a dream come true.

The person who comes next is Dr. Suprita Talnikar, my wife and collaborator who spent hours patiently listening to my half-baked ideas and incomplete proofs, improving papers to submittable conditions and proofreading them. Her help and support in each and every decision throughout my Ph.D career needs a special mention.

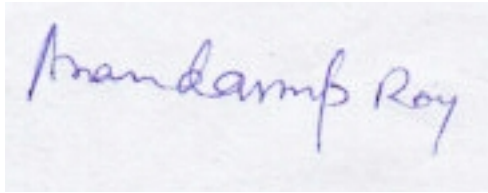
I would also like to express my sincere gratitude to other professors of our unit – Prof. Subhamoy Maitra, Prof. Palash Sarkar and Prof. Kishan Chand Gupta – for their support and useful feedback throughout my research career. My special thanks to my senior Dr. Shion Samadder Chaudhury for his guidance throughout the duration of my Ph.D. I must also deeply thank Professor Sakurai Kouichi for collaborating and guiding me with invaluable information in a very crucial part of my research.

I am also grateful to my colleagues and office mates Dr. Soumya Chattopadhyay, Mr. Chandranan Dhar, Mr. Anik Raychaudhuri, Mr. Rakesh Kumar, Dr. Bishwajit Chakraborty, Mr. Sayantan Paul, Mr. Abishanka Saha, Mr. Arghya Bhattacharjee, Mr. Avishek Majumder, Mr. Samir Kundu and others for our numerous discussion sessions, tea breaks and for their moral support. Thanks should also go to all the staff of the Applied Statistics Unit, ISI Kolkata for their constant help with all official, administrative and other relevant tasks.

I must certainly mention my sister, Arundhati Roy and my innumerable friends, juniors and classmates from ISI Boys Hostel, RS Hostel and M.Tech Hostel, who provided me with much-needed constant moral support and inspiration. Lastly, I would be remiss in not mentioning my parents, Mr. Alokendu Roy and Dr. Shampa Roy for believing in my decision to undertake this huge endeavor and succeed despite numerous roadblocks.

Declaration

I, Anandarup Roy, declare that the work contained in this thesis is original and has not been submitted for the award of any other Degree or Diploma in any other University/Institute.

A rectangular box containing a handwritten signature in blue ink that reads "Anandarup Roy".

19th July 2024

Signature

Date

Abstract

In this thesis, we aim to develop generalised secret sharing protocols to enhance privacy, security and robustness in various applications. We begin by introducing various existing concepts related to secret sharing, including combinatorial repairable threshold schemes (RTSs), ramp schemes, balanced incomplete block designs (BIBDs), frameproofness, verifiability and hierarchy in the access structure.

Our first work, motivated by the concepts of repairable threshold schemes by Stinson et al. develops extendable tensor designs built on balanced incomplete block designs. It then combines this construction with the concepts of frameproofness by Desmedt et al. and consequently presents a frameproof version (which by definition, loses the property of share repairability). This results in a method of generalizing multiple BIBDs into a single, multi-level, ramp-type extendable secret sharing scheme, along with a discussion focusing on improvement of security, and reduction of share size as well as computation, particularly for application in IoT environments. A new graphical approach can be found in our paper that deals with the problem of secret and share reconstruction in the frameproof setup. Furthermore, a generalised combinatorial design resistant to framing has interesting implications in many areas of interest in distributed IoT devices.

Vulnerabilities may arise in communication networks at various stages. For example, at the share distribution stage, anomalies may be introduced during data transfer from the dealer to some players. It is also possible that some (malicious) players try to frame others. Furthermore, there may occur false share contributions by some (malicious) players during the secret reconstruction stage. We present a novel approach to verify correct submission of shares by each participant during secret reconstruction through a lightweight cheater identification algorithm, which significantly improves the computational complexity of verification compared to existing algorithms.

We move on to exploring ramp-type verifiable secret sharing schemes, and the application of hidden access structures in such cryptographic protocols. Inspired by Sehrawat et al.'s access structure hiding scheme, we develop an ϵ -almost access structure hiding scheme, which is verifiable as well as frameproof. We detail how the concept of ϵ -almost hiding is important for incorporating ramp schemes, thus making a fundamental generalisation of this concept. In particular, this proves that tensor designs are verifiable ramp-type secret sharing schemes.

Finally, we explore hierarchy in access structures and formalize our ϵ -almost access structure hiding framework in the context of zero-knowledge proofs. We aim to achieve this by modelling a smart transportation system implemented through a new Hierarchical Secret Sharing (HSS) ramp scheme within this framework and instantiated with ASCON, a good lightweight verification authenticated encryption scheme.

Keywords— combinatorial secret sharing - secure eID - cloud storage - SBIoT - cheater identification - tensor designs - ramp schemes - access structure hiding - verifiability - frameproofness - smart transportation

Related Publications

- (Accepted Paper) Bimal Kumar Roy and Anandarup Roy. Iot-Applicable Generalized Frameproof Combinatorial Designs. IoT, 4(3):466-485, 2023. <https://www.mdpi.com/2624-831X/4/3/20>
- (Accepted Poster) Anandarup Roy, Bimal Kumar Roy, Kouichi Sakurai and Suprita Talnikar. A Combinatorial Approach to IoT Data Security. IWSEC 2023.
- (Submitted Paper) Anandarup Roy, Bimal Kumar Roy, Kouichi Sakurai and Suprita Talnikar. Access Structure Hiding Verifiable Tensor Designs (Submitted to the Journal of Statistics and Applications). IACR Cryptol. ePrint Arch., 902. <https://eprint.iacr.org/2024/902>
- (Ongoing Research) Anandarup Roy, Bimal Kumar Roy, Kouichi Sakurai and Suprita Talnikar. A Secret Sharing Application on a Public Transport Model.

Table of Contents

1	Introduction	1
1.1	Secret Sharing in the Internet of Things	4
1.1.1	Combinatorial RTS	5
1.1.2	Frameproofness	5
1.2	Lightweight Verifiability Through a Combinatorial Approach	6
1.2.1	Vulnerabilities in Communication Networks	7
1.3	Access Structure Hiding Verifiable Tensor Designs	8
1.4	A Secret Sharing Application on a Public Transport Model	10
1.4.1	Securely Updating a Ledger	10
1.4.2	Implementation	11
2	Mathematical Preliminaries	12
2.1	Combinatorial Designs	12
2.2	Matroids and Framing	14
2.3	Graph Theory	20
2.4	Entropy	21
2.5	Interpolation Techniques	22
2.6	Block Ciphers and Authenticated Encryption	28
2.6.1	ASCON	30
3	IoT-Applicable Generalized Frameproof Combinatorial Designs	35
3.1	Introduction	35
3.1.1	Combinatorial RTS	37
3.1.2	A Drawback and An Idea of Extension	37
3.1.3	Frameproofness	38
3.2	Results	38
3.3	Stinson and Wei’s Model	39
3.4	Tensor Design Generated by Two BIBDs	41
3.4.1	Definition of the Krönecker Product	41
3.4.2	Krönecker Product of Two BIBDs	42

3.4.3	Some Results on the Krönecker Product of BIBDs	44
3.4.4	Proof of Existence of Secret Reconstruction	47
3.4.5	A Generalized Share Distribution Scheme	49
3.5	Example	50
3.5.1	Secret Reconstruction	51
3.6	Share Repair for a Krönecker Product-Induced Distribution Design	52
3.7	Frameproofness	55
3.7.1	A Modified Scheme	56
3.7.2	Example	57
3.7.3	Secret Reconstruction for the Modified Scheme	59
3.8	Graphical Representation and Proof of Existence of Permutations	60
3.9	Conclusions and Future Work	61
4	Applications to IoT and Verifiability	63
4.1	Secret Sharing Schemes and the Internet of Things	63
4.2	Vulnerabilities in Communication Networks	65
4.3	Verifiability in Secret Sharing	66
4.4	Lightweight Share Verification	67
4.5	Existing Verification Protocols	68
4.6	An Improved Cheater Detection Algorithm	70
4.7	Conclusion	71
5	Access Structure Hiding Verifiable Tensor Designs	73
5.1	Introduction	73
5.2	Preliminaries	74
5.3	ϵ -Almost Access Structure Hiding Ramp-Type Tensor Designs	77
5.3.1	Tensor Design	80
5.3.2	Secret Sharing Properties of $\mathcal{A} \otimes \mathcal{B}$	81
5.3.3	Frameproofness	81
5.3.4	Secret Sharing Properties of $\mathcal{F}(\mathcal{A}, \mathcal{B})$	84
5.3.5	Graphical Representation	85
5.3.6	Defining Access Structure Tokens	85
5.4	Main results	86

5.5	Proof of Theorems 5.3 and 5.4	87
5.6	Proof of Theorems 5.5 and 5.6	90
5.7	Applications	90
5.8	Conclusion and Future Work	91
6	A Secret Sharing Application on a Public Transport Model	93
6.1	Introduction	93
6.1.1	An Overview of the Model	93
6.1.2	Communication Flow	95
6.2	Implementation	100
6.3	Conclusion	101
7	Conclusions and Open Issues	103
	Bibliography	105

1 | Introduction

This thesis aims to develop a generalized secret sharing protocol that enhances privacy, security, and robustness for various applications. We begin by introducing foundational concepts in secret sharing, including combinatorial repairable threshold schemes (RTSs), ramp schemes, balanced incomplete block designs (BIBDs), verifiability, and hierarchical access structures.

A secret sharing scheme is a cryptographic technique used to divide a secret into multiple parts, called shares, and distribute these shares among different parties. The secret can only be reconstructed when a sufficient number of shares are combined. Such a scheme has a *threshold*, or a minimum number of shares required to recover the original secret; any set of shares less than the threshold reveals no information about the secret. On the other hand, combining the required number of shares allows for the exact recovery of the original secret. For example, imagine a safe with a combination that needs to be split among three people. No single person should be able to open the safe alone, but any two should be able to. This is a basic concept of a (2-out-of-3) secret sharing scheme. By distributing the secret, secret sharing enhances security by preventing a single point of failure and mitigating the risk of unauthorized access.

Our first contribution builds on repairable threshold schemes by Stinson et al. ([Kacsmar & Stinson, 2019](#)), introducing extendable tensor designs based on BIBDs. We combine this with frameproofness concepts from Desmedt et al. ([Desmedt, Mo, & Slinko, 2021](#)) to create a frameproof version, sacrificing share repairability. This leads to a multi-level, ramp-type extendable secret sharing scheme generalizing multiple BIBDs, with a focus on improving security, reducing share size and computation, particularly for IoT applications. We introduce a novel graphical approach for secret and share reconstruction in the frameproof setup.

We extend Stinson's combinatorial model from finite fields \mathbb{F}_{q^k} to integer rings by constructing distribution designs with integer ring entries, demonstrating a simpler ramp scheme and secret reconstruction method. Addressing the security vulnerability of framing players, we generalize combinatorial RTS and enhance our scheme with frameproofness. We believe our results can be extended to arbitrary numbers of distribution designs and that the Krönecker product of BIBDs can be generalized to t-designs, with corresponding results following. A frameproof modification for the generalized scheme remains an open problem. We demonstrate the broad applicability of our tensor design and verification protocol in various IoT contexts.

The Internet of Things (IoT) encompasses a vast array of critical applications, from extensive networks in healthcare, commerce, and government to indispensable everyday devices. These applications universally demand robust privacy and security for personal data while minimizing computational overhead and energy consumption. A generalized combinatorial design resistant to framing holds immense potential for addressing these challenges in distributed IoT environments.

IoT networks are susceptible to vulnerabilities at various stages. The share distribution phase can introduce anomalies during data transfer or malicious framing attempts. Similarly, the secret reconstruction phase may encounter false share contributions from malicious participants. These threats underscore the critical need for robust and secure Verifiable Secret Sharing (VSS) schemes.

Cheater detection is a fundamental component of VSS, ensuring only authorized shareholders with valid shares can reconstruct the secret. This integrity is essential for protecting sensitive information from malicious manipulation. While existing VSS verification protocols such as those employing homomorphic commitments (Benaloh, 1986), share coherence verification (Harn & Lin, 2009), space-efficient techniques (Cafaro & Pellè, 2018; Cafaro & Pellè, 2014) and consensus mechanisms (Geng, Njilla, & Huang, 2022) offer valuable solutions, further advancements are necessary.

We propose an improved cheater detection algorithm that surpasses traditional hash-based methods in computational efficiency through simple algebraic operations. While reducing storage requirements, this algorithm incurs increased communication overhead. This work lays the foundation for future research to enhance VSS security, efficiency, and practical implementation in IoT applications.

We introduce a novel, computationally efficient cheater identification algorithm to verify the integrity of shares during secret reconstruction. Building upon this, we delve into ramp-type verifiable secret sharing schemes and explore the integration of hidden access structures. Inspired by Sehrawat et al.'s (V. S. Sehrawat, Yeo, & Desmedt, 2021) work, we propose an ϵ -almost access structure hiding scheme that is both verifiable and frameproof. This concept is crucial for incorporating ramp schemes, leading to a fundamental generalization of the original concept. Notably, we demonstrate that our tensor designs are verifiable ramp-type secret sharing schemes.

We further explore the verifiability and frameproofness of access structure hiding ramp-type tensor designs by introducing an ϵ -almost access structure hiding (θ, Θ, ℓ) -ramp tensor design, a significant extension of Sehrawat et al.’s work. Our approach leverages Roy et al.’s concept of extending repairable threshold schemes through tensor products of balanced incomplete block designs to enhance data security and privacy. This generalization strengthens the security and verifiability of secret sharing schemes by providing a mechanism for verifying the correctness of received shares and ensuring accurate reconstruction. Incorporating ramp schemes enhances resilience against malicious attacks and unauthorized access.

While our ϵ -almost access structure hiding concept is demonstrated for extendable combinatorial tensor designs, its potential applications extend to various ramp-type schemes, offering opportunities for improved confidentiality, secrecy, and verifiability. We envision practical applications of our techniques in domains demanding robust security, such as secure data sharing, access control, and distributed systems.

To bridge theory and practice, we introduce a novel hierarchical secret sharing (HSS) scheme inspired by (Tassa, 2007), tailored to protect passenger data and travel histories within a smart public transportation system. This system involves multiple entities with varying access requirements. Our HSS scheme, built upon the foundation of tensor-based designs from Chapter 3, offers a hierarchical structure with strong security and efficiency guarantees. By incorporating lightweight cryptographic primitives like ASCON, we address potential vulnerabilities such as communication errors and malicious attacks. This practical implementation demonstrates the feasibility and effectiveness of our ϵ -almost access structure hiding framework in a real-world setting, providing a robust solution for safeguarding sensitive data while maintaining system functionality.

Our approach leverages the strengths of the theoretical concepts discussed in this thesis in verifiable secret sharing, frameproofness, and tensor-based designs; by combining these concepts with hierarchical access control and lightweight cryptography, we create a comprehensive framework that addresses the specific challenges of a smart public transportation system. This work therefore represents a significant step towards realizing secure and efficient data management solutions for complex real-world applications.

1.1 Secret Sharing in the Internet of Things

A secret sharing scheme is a useful tool in modern cryptography. They are distinctive in distributing a secret amongst multiple devices, ensuring that no single device has access to the entire secret. This makes secret sharing schemes ideal for IoT applications where multiple devices need to work together to perform a task. For example, in a smart home system, multiple devices such as sensors, cameras, and smart locks need to communicate with each other to provide security and convenience to the homeowner. In secret sharing-based IoT (SBIoT), each cloud server is given a share constructed using a secret sharing scheme. A collection of servers can reconstruct the secret provided that they satisfy the reconstruction criteria of the underlying scheme (instead of privately owned keys in encryption-based schemes). Such a scheme enables processing without the need of decryption. Energy efficiency refers to the total energy consumption of an IoT network, which affects the lifetime of a network (Shivhare, Maurya, Sarif, & Kumar, 2022). It is well-known that use of a ramp-type scheme improves the security and energy efficiency in SBIoT networks (Tang, 2021). It provides better security against various types of attacks, including replay attack, modification attack, selective forwarding attack, and data leakages when a passive attacker is encountered. These benefits contribute to enhancing the overall security and performance of data transfer in SBIoT networks. Using a threshold scheme enhances personal information protection for eID cards by not storing any personal information per se in the card (Park & Lee, 2018). Instead, sensitive personal information is divided into two parts for distributed storage in the client and the eID card. This ensures safety even when eID cards are lost because none of the original information can be figured out from a single secret share. With this structure, no information whatsoever on the original can be known from only the secret share in the card.

Secret sharing schemes also play a crucial role in ensuring secure data storage within cloud environments. These schemes involve the division of data into multiple shares, which are then stored on different servers. This approach provides a safeguard against any potential compromise of a single server, thereby maintaining the security of the data. In (Nirmala, Bhanu, Patel, & Pvt, 2012), the authors present an exploration of the comparative performance of Shamir's secret sharing algorithm (Shamir, 1979) and Rabin's IDA (Rabin, 1989) within a private cloud framework utilizing the OpenStack cloud infrastructure. The experimental results indicated that Shamir's secret sharing algorithm outperformed Rabin's IDA in terms of generating

the shares and reconstructing the data. However, Rabin’s IDA exhibited a lower storage overhead when compared to Shamir’s secret sharing algorithm. These findings underscore the importance of considering various factors, such as generation time, reconstruction time, and storage requirements, when selecting an appropriate secret sharing scheme for secure data storage in cloud environments.

In (Kacsmar & Stinson, 2019), Stinson and Kacsmar showcased imperfect secret sharing methods that evolved from an ideal scheme like the Shamir scheme. They introduced a threshold scheme that could rebuild lost shares with a certain chance, and protect against adversaries with fewer players than the threshold. Our research expands upon this by broadening the scope of distribution designs, simplifying secret reconstruction and share restoration, and enhancing security across multiple scenarios. In short, we revisit the combinatorial design and some of its key properties first.

1.1.1 Combinatorial RTS

A *repairable threshold scheme (RTS)* is a (τ, b) -threshold scheme in which a subset of players can repair another player’s share in the event that their share is lost or corrupted, without the participation of the dealer who set up the scheme. Stinson and Wei (Stinson & Wei, 2018) introduced *combinatorial RTS* in which, the repairing protocol does not compromise the (unconditional) security of the threshold scheme.

In this work, we generalize the domain by proposing a method to construct a distribution design with entries from an integer ring, and show that this is a ramp scheme. The size of the authorized coalition that can recover the secret is significantly reduced in our framework. Furthermore, the scheme proposed in this thesis produces a far more efficient share repairability, which is possible due to the generalized domain, and based heavily on the easier secret reconstruction mentioned beforehand.

1.1.2 Frameproofness

The concept of frameproofness was examined by Desmedt et al. in their recent paper (Desmedt et al., 2021). Framing a player (or players) clearly compromises the security of any secret sharing system by allowing a group of players to gain unauthorized access to additional information about the secret. Therefore, it is vital to restrict these capabilities and/or the size of any such

group when designing a combinatorial secret sharing scheme. We address this issue in our proposed method and introduce a modified scheme to address the issue of a small coalition size. Specifically, we enhance the extension scheme to prevent any framing by a coalition smaller than the threshold. The question of the minimum coalition size that can frame a player under this new modification remains unanswered.

1.2 Lightweight Verifiability Through a Combinatorial Approach

Secret sharing schemes, particularly in the context of the Internet of Things (IoT), have several important applications that enhance security and data integrity. In IoT networks, devices often need to communicate securely. Protecting sensitive data collected by these devices, such as health data from wearable devices or environmental data from sensors can be encrypted using a secret key derived from a secret sharing scheme. The encrypted data is then distributed among multiple nodes. Only a subset of nodes can collaborate to decrypt the data, ensuring that unauthorized nodes cannot access the original data. Storing data across multiple IoT devices enhances the reliability and security of the system, since the original data can only be reconstructed when a sufficient number of nodes collaborate. This approach also mitigates the risk of data loss due to node failures or attacks. For applications in secure multi-party computation, i.e. enabling multiple IoT devices to perform computations on shared data without revealing the data to each other, secret sharing allows devices to hold shares of inputs and perform computations on these shares. The results can be shared among the devices, and only the final output is revealed, ensuring that individual inputs remain confidential.

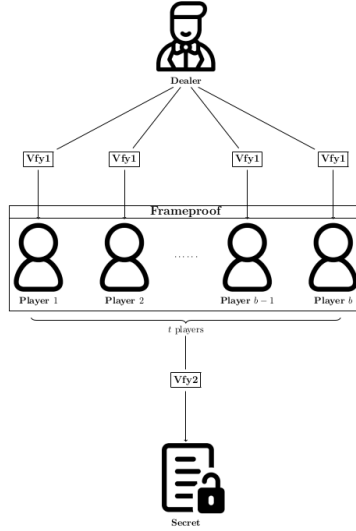
Furthermore, secret sharing can be used to create a multi-factor authentication system where a secret is shared among several devices. A device must present its share along with other factors (like biometric data) to gain access. This adds an additional layer of security, thus enabling authentication and access control. Firmware can also be encrypted using a secret key shared among trusted nodes. The update process can require multiple nodes to verify the integrity of the firmware before it is applied, preventing unauthorized or malicious updates. These are only some examples where secret sharing schemes provide a robust framework for enhancing security in IoT applications by mainly ensuring that sensitive information is distributed, making

it resilient against various types of attacks while maintaining confidentiality and integrity.

Verifiable Secret Sharing (VSS) is a cryptographic protocol that allows a secret to be distributed among a group of participants. In such a scheme, the secret (such as a cryptographic key) is divided into multiple shares, which are distributed to participants. Only a specific subset of these participants (defined by a threshold) can reconstruct the original secret. Each participant can verify that the share they received is correct. This is crucial because it prevents participants from accepting incorrect or tampered shares. In a VSS scheme, additional verification data is provided alongside the shares, allowing participants to check the validity of their shares without needing to communicate with others. VSS schemes are designed to be secure against certain types of attacks, such as collusion among participants. They ensure that even if some participants act maliciously, they cannot reconstruct the secret unless they meet the threshold requirement. VSS is particularly useful in distributed systems, such as in the Internet of Things (IoT), where secure and reliable group communication is essential. It can be used for secure key management, authentication, and other cryptographic applications. Thus, VSS enhances traditional secret sharing by adding a layer of verifiability, ensuring that participants can trust the shares they receive and that the secret can be reconstructed securely.

1.2.1 Vulnerabilities in Communication Networks

- **Share Distribution Stage:** Introduction of anomalies during data transfer from dealer to player
- **Framing Dynamics:** Risks of players framing each other
- **Malicious Share Insertion:** Threats of false share contributions during secret reconstruction



We shall discuss a lightweight share verification protocol for which, the residue computation is at most $\mathcal{O}(\log^2 n)$, and the summation is $\mathcal{O}(n)$.

1.3 Access Structure Hiding Verifiable Tensor Designs

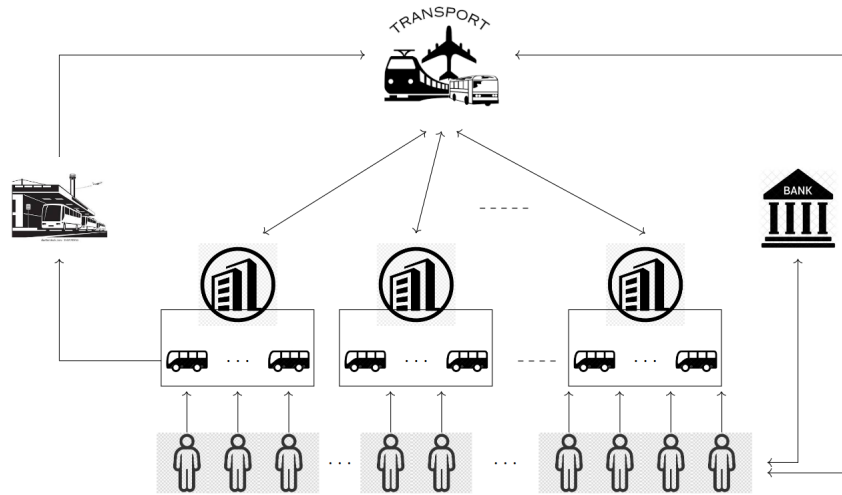
A verifiable secret sharing scheme (Verheul & van Tilborg, 1997; Peng, 2012; Feldman, 1987; Pedersen, 1991; Dehkordi, Farahi, & Mashhadi, 2024) is a cryptographic protocol that allows a dealer to distribute shares of a secret to a group of parties in such a way that (i) the secret remains confidential and cannot be determined by any unauthorized collection of parties, (ii) the secret can be reconstructed correctly by the authorized collection of parties when they combine their shares, (iii) there is a mechanism for parties to verify the correctness of the shares they receive and for the reconstruction process, and (iv) the scheme can withstand malicious behavior from both the dealer and the parties, thus ensuring the security and integrity of the secret sharing process.

Repairable Threshold Schemes (RTSs) (Stinson & Wei, 2018; Laing & Stinson, 2018) are cryptographic schemes that allow for the reconstruction of lost or corrupted shares in a threshold scheme without the need for the dealer who initially set up the scheme to be involved in the repair process. In RTSs, a subset of authorized parties can collaboratively reconstruct the lost share, ensuring the integrity and availability of the shared secret. Chapter 3 discusses how (B. K. Roy & Roy, 2023) explores the concept of repairable ramp schemes for secret sharing and various applications, including cloud storage, sensor-based IoTs, and electronic identification cards.

It proposes a protocol for extending schemes that allow for the retrieval of shares through collaborative efforts in case of loss or corruption, thereby enhancing data security and privacy. It also introduces the concept of tensor products of balanced incomplete block designs (BIBDs), which help securely combine individual secrets from various systems, enabling multi-level or multi-system secret sharing schemes in a robust and efficient manner. (Desmedt et al., 2021) introduced the concept of frameproofness of secret sharing schemes, which ensures the security and integrity of shared secrets and analyses the resistance of a scheme to attempts of falsely implicating (framing) a (set of) player(s) in the unauthorized disclosure of secret information. 3 establishes a theoretical framework for frameproofness within its extension protocol, and ensures that its extended scheme upholds the principles of frameproofness by leveraging concepts from combinatorial design theory.

(V. S. Sehrawat et al., 2021) provide a detailed discussion on how secret sharing can be achieved with access structures hidden from any unauthorized coalition of players, allowing for a wide range of access policies to be enforced in the secret sharing process. The scheme is designed to support verifiability even when a majority of the parties are malicious, and its verification procedure does not incur any communication overhead, making it “free” in terms of computational resources. The scheme provides a maximum share size formula that allows for efficient sharing of secrets while maintaining security guarantees. The share size is optimized to balance security and efficiency considerations. It also includes mechanisms to detect and identify malicious behavior during the secret sharing process.

In this thesis, we introduce a novel framework for ϵ -almost access structure hiding ramp-type tensor designs. Building upon existing work on secret sharing, including VSS, RTSs, BIBDs, and access structure hiding schemes, we formalize our proposed ϵ -almost access structure hiding scheme. By extending the concept of tensor designs and incorporating frameproofness, we establish a robust secret sharing mechanism with provable security properties. Our contributions include the development of an efficient access structure token generation algorithm and rigorous proofs of the scheme’s correctness, secrecy, and verifiability. Several suggestions about practical applicability in real-world use-cases are also listed.



1.4 A Secret Sharing Application on a Public Transport Model

Next, we shall formalize the ϵ -almost access structure hiding framework in the context of zero-knowledge proofs by introducing a new ramp-type hierarchical secret sharing scheme motivated by (Dutta, Paul, Ozaki, Ranzan, & Sakurai, 2021) of Dutta et al., within this framework. We shall discuss applications of this scheme in IoT as well as other use-cases such as in ledger management situations. Finally, we shall also describe a verification protocol through a good lightweight authenticated encryption scheme, say Ascon (Dobraunig, Eichlseder, Mendel, & Schl affer, 2021).

1.4.1 Securely Updating a Ledger

We assume that every passenger has a travel id card and a bank account. These are some questions we attempt to answer with our framework:

- Every bus maintains a ledger. With whom does it communicate this?
- How to encrypt this bus ledger?
- How to consolidate it with the station ledger?
- What can be a good ledger updating protocol for \mathcal{E} ?
- Who can read the ledger(s)?
- In which communication channels can errors occur?

- Which communication channels can be affected by malicious entities?
- Which participants can be affected by framing attacks from other participants?
- How to ensure secure communication (verifiability)?
- How to protect the participants from framing and other attacks?
- How to ensure that our system is lightweight, secures all passenger data, and satisfies all the above requirements?

1.4.2 Implementation

Our goal is to protect the confidential travel information and payment records of passengers from potential threats (such as the bus companies). To achieve this, it is necessary to integrate access structure tokens and (ϵ -almost) access structure hiding into a practical and secure secret sharing system that is resistant to tampering. Additionally, we need to clearly identify which parties could be considered malicious or simply potential sources of errors in the communication network, and develop secure and efficient algorithms for verification and error correction to be utilized in the interactions within this system. We suggest the adoption of the lightweight authenticated encryption scheme ASCON for the verification aspect of our proposed model.

Before studying the above questions, we shall study some of the properties of the answers. We begin in Chapter 2 with some definitions and basic Mathematical preliminaries.

2 | Mathematical Preliminaries

This section briefly describes some concepts required to read further. An interested reader is requested to refer the books or the papers associated for the proof of the theorems and results mentioned here.

2.1 Combinatorial Designs

Combinatorial design theory concerns questions about whether it is possible to arrange elements of a finite set into subsets so that certain “balance” properties are satisfied. We recall some basic definitions and properties of certain types of designs from (Stinson, 2004). Firstly, given a set of points X and a collection (i.e., multiset) \mathcal{A} of nonempty subsets of X called blocks, the pair (X, \mathcal{A}) is a design.

Definition 2.1 *Let v, k and λ be positive integers such that $v > k \geq 2$. A (v, k, λ) -balanced incomplete block design (which we abbreviate to (v, k, λ) -BIBD) is a design (X, \mathcal{A}) such that the following properties are satisfied:*

1. $|X| = v$,
2. each block contains exactly k points, and,
3. every pair of distinct points is contained in exactly λ blocks.

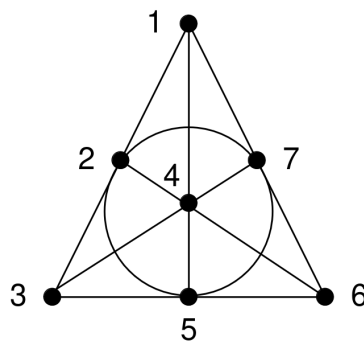


Figure 2.1: The Fano Plane – An example of a $(7, 3, 1)$ -BIBD with $X = \{1, 2, 3, 4, 5, 6, 7\}$ and $\mathcal{A} = \{123, 145, 167, 246, 257, 347, 356\}$.

The following are some properties of balanced incomplete block designs.

Theorem 2.1 *In a (v, k, λ) -BIBD, every point occurs in exactly $r := \frac{\lambda(v-1)}{k-1}$ blocks.*

Theorem 2.2 A (v, k, λ) -BIBD has exactly $b := \frac{vr}{k} = \frac{\lambda(v^2 - v)}{k^2 - k}$ blocks.

Corollary 2.1 If a (v, k, λ) -BIBD exists, then $\lambda(v - 1) \equiv 0 \pmod{k - 1}$ and $\lambda v(v - 1) \equiv 0 \pmod{k(k - 1)}$.

Sometimes we shall use the notation (v, b, r, k, λ) -BIBD if we want to record the values of all five parameters.

Definition 2.2 Let (X, \mathcal{A}) be a design where $X = \{x_1, \dots, x_v\}$ and $\mathcal{A} = \{A_1, \dots, A_b\}$. The incidence matrix of (X, \mathcal{A}) is the $v \times b$ 0-1 matrix $M = (m_{i,j})$ defined by the rule

$$m_{i,j} = \begin{cases} 1 & \text{if } x_i \in A_j; \\ 0 & \text{if } x_i \notin A_j. \end{cases}$$

Suppose that (X, \mathcal{A}) is a design with $|X| = v$ and $|\mathcal{A}| = b$. Let M be the $v \times b$ incidence matrix of (X, \mathcal{A}) . The design having incidence matrix M^\top is called the dual design of (X, \mathcal{A}) . Suppose that (Y, \mathcal{B}) is the dual design of (X, \mathcal{A}) ; then $|Y| = |\mathcal{A}| = b$ and $|\mathcal{B}| = |X| = v$. Properties of dual designs of BIBDs are summarized in the following theorem.

Theorem 2.3 Suppose that (X, \mathcal{A}) is a (v, b, r, k, λ) -BIBD, and let (Y, \mathcal{B}) be the dual design of (X, \mathcal{A}) . Then the following properties hold:

1. every block in \mathcal{B} has size r ,
2. every point in Y occurs in exactly k blocks in \mathcal{B} , and
3. any two distinct blocks $B_i, B_j \in \mathcal{B}$ intersect in exactly λ points.

Theorem 2.4 (Fisher's Inequality) In any (v, b, r, k, λ) -BIBD, $b \geq v$.

Note that t -designs generalise BIBDs.

Definition 2.3 Let v, k, λ , and t be positive integers such that $v > k \geq t$. A t - (v, k, λ) -design is a design (X, \mathcal{A}) such that the following properties are satisfied:

1. $|X| = v$,
2. each block contains exactly k points, and

3. every set of t distinct points is contained in exactly λ blocks.

The general term t -design is used to indicate any t - (v, k, λ) -design.

Theorem 2.5 Suppose that (X, \mathcal{A}) is a t - (v, k, λ) -design. Also suppose that $Y \subseteq X$, where $|Y| = s \leq t$. Then there are exactly

$$\lambda_s := \frac{\lambda \binom{v-s}{t-s}}{\binom{k-s}{t-s}}$$

blocks in \mathcal{A} that contain all the points in Y .

Corollary 2.2 Suppose that (X, \mathcal{A}) is a t - (v, k, λ) -design, and $1 \leq s \leq t$. Then (X, \mathcal{A}) is an s - (v, k, λ_s) -design, where

$$\lambda_s = \frac{\lambda \binom{v-s}{t-s}}{\binom{k-s}{t-s}}.$$

Theorem 2.6 For all positive integers t, k and v such that $t < k < v - t$, there exists a nontrivial t - (v, k, λ) -design for some positive integer λ .

2.2 Matroids and Framing

(Desmedt et al., 2021) provides a detailed overview of the important concepts in secret sharing. Furthermore, they also introduce the new concept of *framing* in various cases. We shall summarize these here in short.

A secret sharing scheme contains a set $P = \{p_0, p_1, p_2, \dots, p_n\}$ of participants, where p_0 is designated as the dealer. The dealer is responsible for distributing the secret among the participants. The set \mathcal{K} is defined as the set of all possible secrets, while S_i denotes the set of all possible shares for participant p_i . The shares are the pieces of information that participants receive, which collectively allow them to reconstruct the secret. A distribution table $T \subseteq \mathcal{K} \times S_1 \times S_2 \times \dots \times S_n$ represents the possible distributions of secrets and shares among participants. When a secret $s_0 \in \mathcal{K}$ is to be distributed, an n -tuple $(s_0, s_1, s_2, \dots, s_n) \in T$ is chosen uniformly at random, where each s_i is the share given to participant p_i . The dealer always knows the secret, ensuring that the distribution process is controlled and secure. The set of possible shares (of all participants) is denoted by \mathcal{S} .

Definition 2.4 The information rate ρ of a threshold scheme is the ratio of the size of the secret to the size of a player's share, i.e.

$$\rho = \frac{\log_2 |\mathcal{K}|}{\log_2 |\mathcal{S}|},$$

where \mathcal{S} is the set of all possible shares and \mathcal{K} is the set of all possible secrets.

The distribution table can be represented as a matrix M with entries from the union of the sets of secrets and shares, specifically $\mathcal{K} \cup S_1 \cup \dots \cup S_n$. The rows of the matrix correspond to different methods of distributing the secret, while the columns correspond to the participants in the secret sharing scheme. An arbitrary element of the distribution table is denoted as $M(r, p)$, where r is the index of the row corresponding to a specific method of distribution, and p is the index of a participant in the set P . The distribution table M is considered public knowledge, meaning that all participants have access to this information. This transparency is crucial for the functioning of the secret sharing scheme, as it allows participants to understand the structure of the shares they receive. For a subset $A \subseteq P$, the notation $M(r, A)$ is used to denote the row r of the matrix restricted to the columns corresponding to the participants in A . This allows for analysis of the information available to specific coalitions of participants.

If a coalition A has no information about the share of another participant b (denoted as $A \not\rightarrow b$), it means that for any row r and any possible share $s \in S(b)$, there exists another row r' such that $M(r, A) = M(r', A)$ and $M(r', b) = s$. Conversely, if A knows the share given to b – denoted as $A \Rightarrow b$, which occurs when $M(r, A) = M(r', A) \implies M(r, b) = M(r', b)$ – it implies that the knowledge of A about the distribution allows them to determine the value of $M(r, b)$. This mathematical framework of the distribution table is essential for analyzing the properties and security of secret sharing schemes, particularly in the context of framing and seniority.

Definition 2.5 A coalition $A \subseteq P$ is said to be authorized if it can reconstruct the secret.

This means that the coalition A belongs to the access structure \mathcal{A} , denoted as $A \in \mathcal{A}$. The access structure is a collection of all authorized coalitions that can access the secret. Formally:

Definition 2.6 An access structure \mathcal{A} is defined as a collection of subsets of participants P such that if a coalition X is in \mathcal{A} and Y is a superset of X (i.e., $Y \supseteq X$), then Y is also in \mathcal{A} . Formally, this is expressed as: If $X \in \mathcal{A}$ and $Y \supseteq X$, then $Y \in \mathcal{A}$. This property is known as the monotonicity property of access structures.

Definition 2.7 *Authorized coalitions of the minimum size (i.e. no coalition of smaller size can be authorized) are called minimal authorized coalitions.*

There may be multiple minimal authorized coalitions for an access structure.

Definition 2.8 *The minimal access structure \mathcal{A}_{min} consists of all minimal authorized coalitions. A coalition X is considered minimal if it is authorized, but removing any participant from X results in a coalition that is no longer authorized. Formally:*

$$\mathcal{A}_{min} = \{X \subseteq P \mid |X| = t \text{ and } X \text{ is authorized}\}.$$

For example, in Shamir's t -out-of- n scheme, the minimal access structure is defined as: $\mathcal{A}_{min} = \{X \subseteq P \mid |X| = t\}$. Clearly, access structures – and especially minimal access structures – are a fundamental characteristic of a secret sharing scheme; knowing the full access structure means knowing the secret sharing scheme, and it can therefore be used to characterise the scheme.

We now take a look at certain essential properties of secret sharing schemes.

Definition 2.9 *A secret sharing scheme is said to be connected if every participant $p \in P$ is contained in at least one minimal authorized coalition. This means that no participant is a "dummy" (i.e., a participant that does not contribute to any authorized coalition). Formally, for every participant p , there exists a coalition $X \in \mathcal{A}_{min}$ such that $p \in X$.*

Definition 2.10 *A secret sharing scheme is defined as perfect if any coalition $A \subseteq P$ that does not know the secret has no information about it. This can be expressed mathematically as: $A \rightarrow p_0 \implies A \Rightarrow p_0$, where p_0 is the dealer. In other words, if coalition A does not have access to the secret, it cannot infer any information about it.*

Definition 2.11 *A secret sharing scheme is considered ideal if it is perfect and the cardinality of the set of secrets $|\mathcal{K}|$ is equal to the cardinality of each set of possible shares $|S_i|$ for all participants i . This means: $|\mathcal{K}| = |S_1| = |S_2| = \dots = |S_n| = q$, where q is the size of the finite field from which the secrets and shares are drawn. An ideal secret sharing scheme is thus a perfect scheme with the shortest possible shares.*

Having reviewed the fundamentals of secret sharing schemes, we now turn our attention to

a combinatorial structure known as matroids. As we shall subsequently demonstrate (in say, Theorem 2.7), matroids can be employed to define access structures for secret sharing schemes.

Definition 2.12 *A matroid is a combinatorial structure that generalizes the notion of linear independence in vector spaces. Formally, a matroid M is defined as a pair (E, I) , where:*

- E is a finite set, called the ground set. - I is a collection of subsets of E that satisfies the following properties:

1. **Empty Set** *The empty set is in I : $\emptyset \in I$.*
2. **Hereditary Property** *If a set X is in I and Y is a subset of X , then Y is also in I :
If $X \in I$ and $Y \subseteq X$, then $Y \in I$.*
3. **Exchange Property** *If X and Y are in I and $|X| > |Y|$, then there exists an element $x \in X \setminus Y$ such that $Y \cup \{x\}$ is also in I : If $X, Y \in I$ and $|X| > |Y|$, then $\exists x \in X \setminus Y$ such that $Y \cup \{x\} \in I$.*

A set $X \subseteq E$ is called independent if $X \in I$; otherwise, it is called dependent. A set X is a circuit if it is minimally dependent, meaning that it is dependent but any proper subset of it is independent.

A matroid port is a specific subset of a matroid that retains certain properties of the original matroid. Formally, given a matroid $M = (E, I)$ and an element $p \in E$, the matroid port of M at point p , denoted as $\mathcal{P}_p(M)$, is defined as the set of subsets $X \subseteq E \setminus \{p\}$ such that the rank of X is equal to the rank of $X \cup \{p\}$. Mathematically, this can be expressed as: $\mathcal{P}_p(M) = \{X \subseteq E \setminus \{p\} \mid r(X \cup \{p\}) = r(X)\}$, where $r(X)$ denotes the rank of the set X .

In simpler terms, a matroid port captures the idea of how the inclusion of a specific element p affects the independence of subsets of the remaining elements. If $\mathcal{P}_p(M)$ is an access structure, it indicates that the structure of the matroid is preserved when considering the independence of subsets excluding the element p .

Theorem 2.7 (Brickell-Davenport) *Let M be a connected ideal secret sharing scheme on a set of participants P . Then the sets $D(M) = \{A \subseteq P \mid \exists y \in A \text{ such that } A \setminus y \rightarrow y\}$ are the dependent sets of a connected matroid.*

A coalition $C \subseteq P$ can frame a participant $p \notin C$ if the following condition holds: C can

compute the share of p from their own shares. Mathematically, this can be expressed as:

$$C \text{ can frame } p \iff \exists S \subseteq C \text{ such that } S \text{ can reconstruct } p's \text{ share } s_p.$$

Next, we consider two concepts fundamental to the consideration of framing in secret sharing:

Definition 2.13 *A participant's seniority is defined in the context of the access structure, where a higher seniority implies a greater ability to influence the framing process. A participant p' is considered essential for coalition X if removing p' from X would prevent X from being able to reconstruct the secret.*

(Desmedt et al., 2021) presents a key theorem regarding framing in ideal access structures:

Theorem 2.8 *In an ideal secret sharing scheme, an authorized coalition X can frame a participant p if and only if:*

- p is at least as senior as at least one member of X , and
- X contains a participant p' that is essential for the coalition X .

This theorem establishes a critical relationship between the seniority of participants and the ability of coalitions to frame others. The concepts of seniority and essentiality of a participant are crucial in determining the framing capabilities of coalitions. The theorem implies that if an authorized coalition contains a pivotal member who is essential, they can compute the share of a participant who is at least as senior as that member. This creates a potential vulnerability where participants can be framed based on their seniority and the structure of the coalition. The implications of framing in ideal access structures are significant for the design and security of secret sharing schemes. If authorized coalitions can frame participants, it raises concerns about the integrity of the shares and the potential for misuse. This understanding emphasizes the need for careful design of access structures to mitigate the risks associated with framing, ensuring that participants' shares remain confidential and secure.

Hierarchical Secret Sharing:

A hierarchical secret sharing (HSS) scheme is one whose access structure is partitioned into two or more subcollections, with an order (i.e. a *hierarchy*) defined on the partitions such that

shares from higher priority partitions are “more important” for secret reconstruction. An access structure Γ is said to be hierarchical if it is defined by a seniority relation \succ that is a non-strict linear order. This means that for any two participants p_i and p_j in the set of participants P , we can determine if one is at least as senior as the other, and this relation has no cycles. Let $P = \{p_0, p_1, p_2, \dots, p_n\}$ be the set of participants, where p_0 is the dealer (the most senior participant). The seniority relation \succ implies that if $p_i \succ p_j$, then p_i is no less senior than p_j .

Theorem 2.9 *In a hierarchical access structure, an authorized coalition X can frame a participant p if: - p is at least as senior as the least senior member of X , and - X contains a participant p' that is essential for the coalition X .*

Thus, the seniority of participants plays a crucial role in determining the framing capabilities of coalitions in HSS schemes. A participant p can be framed if they are at least as senior as the least senior member of the coalition X . A participant p' is essential for coalition X if removing p' from X would prevent X from being able to reconstruct the secret. This condition is vital for framing capability.

The implications of framing in hierarchical access structures are significant for the design and security of secret sharing schemes. If authorized coalitions can frame participants based on their seniority, it raises concerns about the integrity of the shares and the potential for misuse. This understanding emphasizes the need for careful design of hierarchical access structures to mitigate the risks associated with framing, ensuring that participants' shares remain confidential and secure.

Verifiable Secret Sharing (VSS) Schemes:

Verifiability in secret sharing schemes is a critical property that ensures the integrity and correctness of the shares distributed among participants. It allows participants to confirm that the shares they receive are valid and that they can reconstruct the secret accurately when combining their shares. A secret sharing scheme is said to be verifiable if there exists a verification algorithm that allows each participant to check the validity of their share, and/or if there exists a verification algorithm that allows a collection of players to check the validity of all shares contributed by that collection. Thus, A VSS scheme is one that can withstand active attacks, specifically:

- a dealer sending inconsistent or incorrect shares to some of the participants during the

distribution protocol, and

- participants submitting incorrect shares during the reconstruction protocol.

If the shares of the players involved are valid, then the reconstruction of the secret from the shares must yield the original secret. Mathematically, if $\text{Recon}(s_A) = S$ for a set of participants A that can reconstruct the secret, then the shares must be valid. On the other hand, if a share is invalid, the verification algorithm must output a negative result. This ensures that participants cannot mistakenly believe they have a valid share when they do not. Finally, if a share is valid, the verification algorithm must output a positive result. This guarantees that valid shares are accepted.

2.3 Graph Theory

A graph G is defined as having a vertex set V (or $V(G)$), an edge set E (or $E(G)$), and a function that assigns to each edge $e \in E(G)$ an unordered pair x, y of vertices known as the endpoints (or simply the ends) of e . An edge is considered incident with its ends, and it connects the ends. If the endpoints of an edge are the same ($x = y$), in which case the edge is referred to as a loop. A vertex is said to be isolated when it has no edges incident with it.

Given an undirected graph \mathcal{G} , a *matching* of \mathcal{G} is a collection \mathcal{M} of edges of \mathcal{G} such that no two edges in \mathcal{M} share a vertex. \mathcal{M} is a *maximal matching* of \mathcal{G} if it is not a subset of any other matching of \mathcal{G} . Thus, adding even one more edge to a maximal matching \mathcal{M} ensures that it is no longer a matching. The number of edges in a maximal matching of \mathcal{G} of the largest size is called the *matching number* of \mathcal{G} . A *perfect matching* \mathcal{M} of \mathcal{G} is such that each vertex of \mathcal{M} has an edge incident to it.

A *vertex cover* of a graph \mathcal{G} is a collection of vertices of \mathcal{G} such that every edge is incident to at least one vertex in the collection; an *edge cover* of a graph \mathcal{G} is a collection of edges of \mathcal{G} such that every vertex of \mathcal{G} has at least one edge from the collection incident to it. A *minimal vertex cover* (respectively *minimal edge cover*) is one that is not a proper subset of any other vertex cover (respectively edge cover). Thus, if \mathcal{G} has no isolated vertices, then the sum of the number of vertices in its minimal vertex cover and the number of edges in its minimal edge cover equals the total number of its vertices.

If the vertex set \mathcal{V} of a graph \mathcal{G} can be partitioned into two disjoint subsets as $\mathcal{V} = \mathcal{A} \sqcup \mathcal{B}$ such that any edge from a vertex in \mathcal{A} can only be incident to a vertex in \mathcal{B} and vice versa, then \mathcal{G} is called a *bipartite graph*. Let us recall some interesting results on matching in bipartite graphs.

Theorem 2.10 (König, (König, 1931)) *In any bipartite graph, the number of edges in a maximum matching equals the number of vertices in a minimum vertex cover.*

Theorem 2.11 (Hall, (Hall, 1935)) *Given a bipartite graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $\mathcal{V} = \mathbf{A} \sqcup \mathbf{B}$, \mathcal{G} has a matching of size $|\mathbf{A}|$ if and only if for every $S \subseteq \mathbf{A}$ we have $|N(S)| \geq |S|$, where $N(S) = \{b \in \mathbf{B} : \exists a \in S \text{ with } (a, b) \in \mathcal{E}\}$.*

2.4 Entropy

Information theory is the mathematical study of the quantification, storage, and communication of information. A key measure in information theory is entropy. Entropy quantifies the amount of uncertainty involved in the value of a random variable or the outcome of a random process.

Definition 2.14 *The entropy of a random variable X with probability mass function $p(x)$ is defined by*

$$H(x) := - \sum_{x \in X} p(x) \log_2 p(x) = E_p \left(\log_2 \frac{1}{p(x)} \right).$$

Definition 2.15 *The joint entropy $H(X, Y)$ of a pair of random variables (X, Y) with joint probability mass function $p(x, y)$ is given by the expression*

$$H(X, Y) := - \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log_2 p(x, y).$$

The conditional entropy of Y given X is defined as

$$H(Y|X) := \sum_{x \in \mathcal{X}} p(x) H(Y|X = x).$$

Definition 2.16 *The joint entropy of two random variables can be defined as*

$$H(X, Y) := H(X) + H(Y|X)$$

Definition 2.17 *The mutual information of two random variables is defined by*

$$I(X; Y) := \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \frac{\log p(x, y)}{p(x)p(y)}$$

We shall not dive further into information theory here as not much is required in our work.

2.5 Interpolation Techniques

Interpolation is a fundamental concept in numerical analysis and approximation theory, which involves estimating the values of a function at points that are not explicitly known, based on its values at a set of known points. The goal is to construct a new function that passes through these known points, providing a good approximation of the original function. It is a powerful mathematical tool that allows for the estimation of unknown values based on known data. The choice of interpolation method (Lagrange, Hermite, etc.) depends on the specific requirements of the problem, such as the need for derivative matching or the nature of the data. Understanding the underlying theory and error analysis is crucial for effectively applying interpolation techniques in practical scenarios. Interpolation has numerous applications across various fields, such as for numerical integration and solving differential equations, for rendering curves and surfaces in computer graphics, for reconstructing signals from sampled data in signal processing, and for creating models that approximate real-world data. Interpolation can be broadly categorized into two types: polynomial interpolation involves finding a polynomial that passes through a given set of points, while piecewise interpolation involves constructing a piecewise-defined function that approximates the original function.

Lagrange Interpolation:

The Lagrange interpolation formula provides a way to construct a polynomial that passes through a given set of points. It is a method for constructing a polynomial that passes through a given set of points. It is particularly useful because it provides a straightforward way to find the interpolating polynomial without needing to solve a system of equations.

Definition 2.18 *Given a set of $n + 1$ distinct data points $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$, where*

$y_i = f(x_i)$ for some function f , the Lagrange interpolating polynomial $P(x)$ is defined as:

$$P(x) = \sum_{i=0}^n y_i L_i(x),$$

where $L_i(x)$ are the Lagrange basis polynomials defined as:

$$L_i(x) = \prod_{\substack{0 \leq j \leq n \\ j \neq i}} \frac{x - x_j}{x_i - x_j}.$$

Each $L_i(x)$ is constructed such that $L_i(x_j) = \delta_{ij}$ (the Kronecker delta), meaning $L_i(x)$ is 1 at $x = x_i$ and 0 at all other x_j .

Properties of Lagrange Basis Polynomials:

Degree. Each $L_i(x)$ is a polynomial of degree n .

Interpolation Property. $L_i(x_j) = \delta_{ij}$, meaning $L_i(x)$ is equal to 1 at $x = x_i$ and 0 at all other x_j (for $j \neq i$).

Unique Polynomial. The polynomial $P(x)$ is the unique polynomial of degree at most n that passes through the points $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$.

Construction of the Lagrange Polynomial:

Identify the Data Points. Choose $n + 1$ distinct points $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$.

Calculate the Basis Polynomials. For each i , compute $L_i(x)$ using the formula provided above.

Form the Polynomial. Substitute the values y_i into the polynomial expression:

$$P(x) = \sum_{i=0}^n y_i \prod_{\substack{0 \leq j \leq n \\ j \neq i}} \frac{x - x_j}{x_i - x_j}.$$

The error in Lagrange interpolation can be expressed as:

$$E(x) = f(x) - P(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{i=0}^n (x - x_i),$$

for some ξ in the interval containing the x_i . This shows that the error depends on the $(n + 1)^{\text{th}}$ derivative of the function f and the distance from the interpolation points.

For example, consider the points $(1, 2)$, $(2, 3)$, and $(3, 5)$. We want to find the polynomial $P(x)$ that passes through these points. For the points

$$\begin{aligned}x_0 &= 1 & , & & y_0 &= 2 \\x_1 &= 2 & , & & y_1 &= 3 \\x_2 &= 3 & , & & y_2 &= 5,\end{aligned}$$

we calculate basis polynomials as follows:

$L_0(x)$:

$$L_0(x) = \frac{(x - 2)(x - 3)}{(1 - 2)(1 - 3)} = \frac{(x - 2)(x - 3)}{(-1)(-2)} = \frac{(x - 2)(x - 3)}{2}.$$

$L_1(x)$:

$$L_1(x) = \frac{(x - 1)(x - 3)}{(2 - 1)(2 - 3)} = \frac{(x - 1)(x - 3)}{(1)(-1)} = -(x - 1)(x - 3).$$

$L_2(x)$:

$$L_2(x) = \frac{(x - 1)(x - 2)}{(3 - 1)(3 - 2)} = \frac{(x - 1)(x - 2)}{(2)(1)} = \frac{(x - 1)(x - 2)}{2}.$$

Finally, we can form the polynomial as $P(x) = 2L_0(x) + 3L_1(x) + 5L_2(x)$. Substituting the basis polynomials then gives:

$$P(x) = 2 \cdot \frac{(x - 2)(x - 3)}{2} - 3(x - 1)(x - 3) + 5 \cdot \frac{(x - 1)(x - 2)}{2}.$$

Hermite Interpolation:

Hermite interpolation extends Lagrange interpolation by not only matching the function values but also the derivatives at specified points. Given a set of points and the desired values of the function and its derivatives, the Hermite polynomial can be constructed similarly to Lagrange but incorporates derivative information. It is useful in constructing polynomials that not only fit a set of data points but also respect the derivatives at those points. This makes it particularly useful in applications where the smoothness of the function is important.

Definition 2.19 Given a set of m distinct data points x_0, x_1, \dots, x_m and corresponding function values $f(x_i)$ and derivative values $f'(x_i)$, the Hermite interpolating polynomial $H(x)$ is constructed to satisfy:

$$H(x_i) = f(x_i) \quad \text{and} \quad H'(x_i) = f'(x_i) \quad \text{for } i = 0, 1, \dots, m.$$

Construction of the Hermite Polynomial:

Using Divided Differences. The Hermite polynomial can be expressed as:

$$H(x) = \sum_{i=0}^m (f(x_i)H_i(x) + f'(x_i)H'_i(x)),$$

where $H_i(x)$ is the Lagrange basis polynomial for the i^{th} point, and $H'_i(x)$ is its derivative.

Using Lagrange Basis Polynomials. The Lagrange basis polynomial $H_i(x)$ for Hermite interpolation is defined as:

$$H_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^m \frac{(x - x_j)}{(x_i - x_j)}.$$

For each point x_i , we need to account for the multiplicity of the derivatives. If f has a derivative of order k at x_i , the basis polynomial is modified to:

$$H_i(x) = \frac{(x - x_i)^k}{k!} \prod_{\substack{j=0 \\ j \neq i}}^m \frac{(x - x_j)}{(x_i - x_j)}.$$

The error in Hermite interpolation can be expressed similarly to Lagrange interpolation. If $H(x)$ is the Hermite interpolating polynomial for $f(x)$, the error can be given by:

$$E(x) = f(x) - H(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{i=0}^m (x - x_i)^2,$$

for some ξ in the interval containing the x_i . This shows that the error depends on the $(n+1)^{\text{th}}$ derivative of the function f and the square of the distance from the interpolation points.

For example, consider the function $f(x) = e^x$ and we want to interpolate it at the points $x_0 = 0$

and $x_1 = 1$ with the following conditions:

$$\begin{aligned} f(0) &= 1 & , & & f'(0) &= 1 \\ f(1) &= e & , & & f'(1) &= e \end{aligned}$$

We must first identify the points and derivatives:

$$\begin{aligned} x_0 &= 0, & f(0) &= 1, & f'(0) &= 1 \\ x_1 &= 1, & f(1) &= e, & f'(1) &= e, \end{aligned}$$

and then construct the basis polynomials as follows:

$H_0(x)$:

$$H_0(x) = \frac{(x-0)^2}{2!} \cdot \frac{(x-1)}{(0-1)} = \frac{x^2}{2} \cdot (x-1) = \frac{x^2(x-1)}{2}.$$

$H_1(x)$:

$$H_1(x) = \frac{(x-1)^2}{2!} \cdot \frac{(x-0)}{(1-0)} = \frac{(x-1)^2}{2} \cdot x.$$

Finally, we form the polynomial as $H(x) = f(0)H_0(x) + f'(0)H_0'(x) + f(1)H_1(x) + f'(1)H_1'(x)$.

Substituting the values gives:

$$H(x) = 1 \cdot H_0(x) + 1 \cdot H_0'(x) + e \cdot H_1(x) + e \cdot H_1'(x).$$

Birkhoff Interpolation:

Birkhoff interpolation is a generalization of polynomial interpolation that allows for the specification of both function values and derivative values at given points. This method is particularly useful in scenarios where one needs to ensure that a polynomial not only passes through certain points but also has specific behaviour (derivatives) at those points. Consider a finite set $X \subseteq \mathbb{R}$ of points x_1, x_2, \dots, x_k such that $x_1 < x_2 < \dots < x_k$, a matrix E with entries $e_{i,j}$ ($1 \leq i \leq k$, $0 \leq j \leq \ell$) whose rightmost column is non-zero, the set $I(E)$ defined as $\{(i, j) \mid e_{i,j} = 1\}$ along with the parameter $d := |I(E)|$, and a set C of d real values $\{c_{i,j} \mid (i, j) \in I(E)\}$. Then the Birkhoff interpolation problem that corresponds to the triplet

$\langle X, E, C \rangle$ is the problem of finding a polynomial $P(x) \in \mathbb{R}_{d-1}[x]$ that satisfies the d equalities

$$P^{(j)}(x_i) = c_{i,j}, \quad (i, j) \in I(E) \quad (2.1)$$

The matrix E is called the *interpolation matrix*.

Lagrange and Hermite interpolations can be viewed as specific instances of Birkhoff interpolation. In Lagrange interpolation, the interpolation matrix comprises solely one column, as all data corresponds to the zeroth-order derivative. Hermite interpolation matrices, on the other hand, feature rows (representing interpolation points x_i) that initiate with a sequence of 1s followed by 0s, reflecting the given values at that point in the form $P^{(j)}(x)$, $0 \leq j \leq j_i$, for some $j_i \geq 0$. Unlike Lagrange or Hermite interpolation, which are unconditionally well-posed, the Birkhoff interpolation problem may not always yield a unique solution. The system of equations (2.1) translates into a square linear system of equations $A\vec{x} = \vec{b}$, where the vector of unknowns \vec{x} consists of the coefficients of the desired polynomial P , the matrix A is determined by X and E , and the right-hand side \vec{b} comprises the data in C . The pair $\langle X, E \rangle$ is called *regular* if the resulting matrix A is regular, ensuring a unique solution to the system (2.1) for any choice of C ; otherwise it is called singular. The matrix E is termed *regular* or *poised* if $\langle X, E \rangle$ is regular for all $X = \{x_1 < x_2 < \dots < x_k\} \subset \mathbb{R}$.

The subsequent lemma provides a simple necessary condition that E must satisfy, to prevent $\langle X, E \rangle$ from being singular for all X .

Lemma 2.1 (Pòlya's Condition.) *A necessary condition for the well-posedness of the Birkhoff interpolation problem with the interpolation matrix the interpolation matrix E is that for each derivative order t ($0 \leq t \leq \ell$, where ℓ denotes the highest derivative order in the data), there must exist at least $t + 1$ given values of derivatives of the polynomial P of order less than or equal to t . Formally, this condition can be expressed as:*

$$|\{(i, j) \in I(E) \mid j \leq t\}| \geq t + 1, \quad 0 \leq t \leq \ell.$$

While any interpolation problem where one is given at each point, a succession of derivatives ($f, f', f'', f^{(3)}$ etc.) is Hermite interpolation, Birkhoff interpolation is the case of unstructured data, in which only some values of the function or its derivatives may be available at certain

points (for example, one may know the values of f and $f^{(3)}$, just f'' at another point, and $f^{(4)}$ and $f^{(6)}$ at a third point). Birkhoff interpolation is devoted to this distinction between structured and unstructured data, and to the numerical challenges that such lack of structure imposes. The flexibility of Birkhoff interpolation as a method for constructing polynomials that satisfy both function values and derivative conditions at specified points makes it particularly useful in applications requiring smoothness and specific behaviour of the interpolating function.

2.6 Block Ciphers and Authenticated Encryption

A block cipher is a deterministic algorithm that operates on fixed-length groups of bits, called blocks. Block ciphers are the elementary building blocks of many cryptographic protocols. They are ubiquitous in the storage and exchange of data, where such data is secured and authenticated via encryption.

A block cipher uses blocks as an unvarying transformation. Even a secure block cipher is suitable for the encryption of only a single block of data at a time, using a fixed key. A multitude of modes of operation have been designed to allow their repeated use in a secure way to achieve the security goals of confidentiality and authenticity. However, block ciphers may also feature as building blocks in other cryptographic protocols, such as universal hash functions and pseudorandom number generators.

Definition 2.20 *A block cipher consists of two paired algorithms, one for encryption, E , and the other for decryption, D . Both algorithms accept two inputs: an input block of size n bits and a key of size k bits, and both yield an n -bit output block. The decryption algorithm D is defined to be the inverse function of encryption, i.e., $D = E^{-1}$. More formally, a block cipher is specified by an encryption function*

$$E_K(P) := E(K, P) : \{0, 1\}^k \times \{0, 1\}^n \rightarrow \{0, 1\}^n,$$

$$E_K(P) := E(K, P) : \{0, 1\}^k \times \{0, 1\}^n \rightarrow \{0, 1\}^n,$$

which takes as input a key K , of bit length k (called the key size), and a bit string P , of length n (called the block size), and returns a string C , also of n bits. P is called the plaintext, and C is termed the ciphertext. For each K , the function $E_K(P)$ is required to be an invertible mapping

on $\{0, 1\}^n$. The inverse for E is defined as a function

$$E_K^{-1}(C) := D_K(C) = D(K, C) : \{0, 1\}^k \times \{0, 1\}^n \rightarrow \{0, 1\}^n,$$

$$E_K^{-1}(C) := D_K(C) = D(K, C) : \{0, 1\}^k \times \{0, 1\}^n \rightarrow \{0, 1\}^n,$$

taking a key K and a ciphertext C to return a plaintext value P , such that

$$\forall P : D_K(E_K(P)) = P.$$

For each key K , E_K is a permutation over the set of input blocks. Each key selects one permutation from the set of $2^n!$ possible permutations.

Data encryption standard (DES) (DES, 1979), triple DES (3DES or TDEA) (De Cannière, 2005), advanced encryption standard (AES) (AES, 2001), blowfish (Schneier, 1993), twofish (Schneier, Kelsey, Whiting, Wagner, & Hall, 1998), and RC5 (Rivest, 1994) are some examples of block ciphers.

Authenticated Encryption (AE) is a cryptographic technique that combines the properties of confidentiality and authenticity into a single operation. It ensures that a message is not only kept secret from unauthorized parties but also verifies that the message has not been altered in transit. This dual functionality is crucial in modern cryptographic applications, where both data integrity and confidentiality are paramount. It is a critical component of modern cryptographic systems, providing a robust mechanism for ensuring both confidentiality and integrity of data. By combining encryption and authentication into a single operation, AE simplifies the implementation of secure communication protocols and reduces the risk of vulnerabilities associated with separate implementations of confidentiality and authenticity. An AE has confidentiality if it ensures that the plaintext message is not accessible to unauthorized parties. It is typically achieved through encryption, which transforms plaintext into ciphertext using a secret key. The authenticity of an AE guarantees that the message comes from a legitimate sender and has not been tampered with. This is often achieved through the use of Message Authentication Codes (MACs) or digital signatures.

Let K be a secret key, P be the plaintext message, C be the ciphertext, T be the authentication tag, and A be the associated data. The encryption process takes as input, a plaintext P , a

key K and optionally, some associated data A . The encryption function can be denoted as $(E, T) = \text{AE_Encrypt}(K, A, P)$, where E is the encryption function that produces the ciphertext C and the authentication tag T . On input of a ciphertext C , a key K , an authentication tag T and (optionally) an associated data A , the decryption process either outputs a plaintext P if authentication passes, or an error if authentication fails. The decryption function can be represented as $P = \text{AE_Decrypt}(K, A, C, T)$. If the authentication tag T does not match, the decryption process will return an error, denoted \perp . Authenticated encryption schemes must satisfy several security properties:

Confidentiality. The ciphertext C should not reveal any information about the plaintext P without the key K .

Integrity An adversary should not be able to modify the ciphertext C or the associated data A without detection.

Authenticity Only parties possessing the key K should be able to generate valid ciphertexts and authentication tags.

Several modes of authenticated encryption have been standardized, including

- the Galois/Counter Mode (GCM), which combines the counter mode of encryption with Galois mode of authentication,
- the Counter with CBC-MAC (CCM), which combines counter mode encryption with Cipher Block Chaining (CBC) for authentication,
- the Offset Codebook Mode (OCB), which, provides both encryption and authentication in a single pass, etc.

2.6.1 ASCON

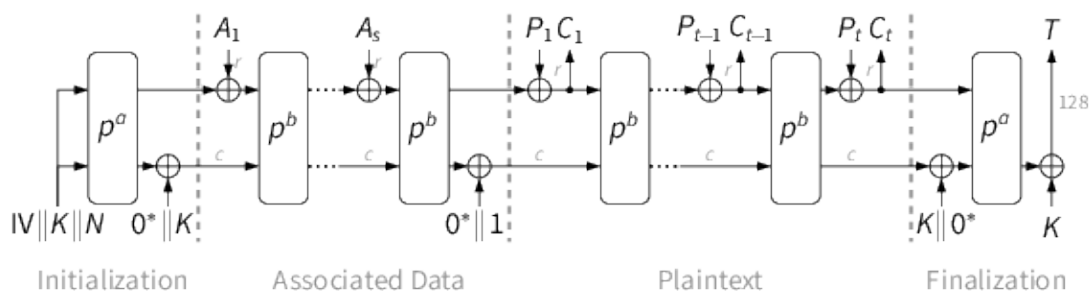
ASCON (Dobraunig et al., 2021) is a lightweight authenticated encryption and hashing scheme designed for resource-constrained environments. It was selected as one of the finalists in the CAESAR competition and is notable for its efficiency and security. It is a robust and efficient authenticated encryption and hashing scheme tailored for lightweight applications. Its sponge-based design, combined with a well-analyzed permutation, provides a strong security foundation while maintaining low resource requirements. ASCON is designed to provide authenticated

encryption with associated data (AEAD) and hashing capabilities. It is particularly suitable for applications in the Internet of Things (IoT) and other environments where computational resources are limited. ASCON is based on the sponge construction, which is a flexible framework for building cryptographic primitives. The key features of ASCON include:

Lightweight Design. Optimized for low memory and computational overhead.

Security Margins. Provides a generous security margin against known cryptanalytic attacks.

Sponge Construction. Utilizes a permutation-based approach that allows for efficient processing of data.



ASCON operates on a state represented as a 320-bit value, divided into five 64-bit lanes:

$$\text{State} = (s_0, s_1, s_2, s_3, s_4) \quad \text{where } s_i \in \mathbb{F}_{2^{64}}.$$

The core of ASCON is a permutation P that transforms the state. The permutation consists of several rounds, each involving substitution and permutation operations. The number of rounds is typically denoted as r . The permutation can be expressed as:

$$\text{State}_{\text{next}} = P(\text{State}_{\text{current}}).$$

ASCON uses a sponge construction for both encryption and hashing. The sponge construction consists of two phases: absorbing and squeezing.

Absorbing Phase. Input data (plaintext, associated data) is absorbed into the state by XORing it with the state and applying the permutation.

Squeezing Phase. The output (ciphertext, hash) is produced by repeatedly applying the permutation and extracting bits from the state.

ASCON provides authenticated encryption through the following algorithms:

The authenticated encryption algorithm $\text{ASCON_AE}(K, N, A, P)$ takes as input, a secret key K (up to 160 bits), a nonce N (128 bits), some associated data A of arbitrary length, and a plaintext P , also of arbitrary length. The output consists of a ciphertext C and an authentication tag T . The encryption process can be summarized as follows:

1. Initialize the state with the key and nonce.
2. Absorb the associated data A .
3. Absorb the plaintext P .
4. Squeeze the output to produce the ciphertext C and authentication tag T .

The decryption algorithm $\text{ASCON_Decrypt}(K, N, A, C, T)$ takes the same inputs as the encryption algorithm, along with the ciphertext C and authentication tag T . The output is the plaintext P or an error if authentication fails. The decryption process involves:

1. Initializing the state with the key and nonce.
2. Absorbing the associated data A .
3. Absorbing the ciphertext C .
4. Verifying the authentication tag T .
5. If verification succeeds, squeezing the output to retrieve the plaintext P .

ASCON also provides a hashing function $\text{ASCON_Hash}(M)$ that operates similarly to the AEAD scheme but focuses on producing a hash output from the input message M .

Security Claims of ASCON:

The ASCON design aims at providing immunity to several possible attacks; the structure of the permutation and the number of rounds are chosen to provide resistance against differential attacks, its design protects against linear cryptanalysis by minimizing linear correlations between the input and output, and the construction ensures collision resistance by making it computationally infeasible to find two distinct inputs that yield the same output. ASCON provides several security claims regarding its authenticated encryption and hashing capabilities. Below are the key security claims along with their corresponding mathematical bounds in terms of the number of bits:

Confidentiality of Plaintext. The confidentiality of the plaintext is guaranteed against chosen plaintext attacks (CPA), with a security of 128 bits against ASCON-128, ASCON-128a and ASCON-80pq.

Integrity of Plaintext. The integrity of the plaintext is ensured, meaning that any modification to the ciphertext will be detected. ASCON also ensures a 128-bit integrity for all three of its variants.

Integrity of Associated Data. The integrity of associated data (A) is also guaranteed, ensuring that any changes to the associated data will be detected. This security is also bound by 128 bits for all three variants.

Integrity of Public Message Number (Nonce). The integrity of the nonce (N) is protected, meaning that any modification to the nonce will be detected. This security is also bound by 128 bits for all three variants.

Key Recovery Resistance. The scheme is designed to resist key recovery attacks, where an attacker attempts to recover the secret key from the ciphertext. The key recovery bounds for the three variants are as follows:

- For ASCON-128: $\min(2^k, 2^{c/2})$ where $k = 128$ and $c = 256$ (resulting in 2^{128} complexity).
- For ASCON-128a: $\min(2^k, 2^{c/2})$ where $k = 128$ and $c = 192$ (resulting in 2^{96} complexity).
- For ASCON-80pq: $\min(2^k, 2^{c/2})$ where $k = 80$ and $c = 128$ (resulting in 2^{64} complexity).

Collision Resistance for Hashing. The hashing function is designed to be collision-resistant, meaning it is computationally infeasible to find two distinct inputs that hash to the same output. This security bound is 128 bits (for fixed output size) in the case of ASCON-hash; for ASCON-Xof, the security is based on the output length, typically providing a security level of $2^{n/2}$ for an output size of n bits.

Security Margin. ASCON has a security margin against known attacks, particularly in its permutation structure. The best attacks against ASCON's initialization have been shown to require complexity significantly below 2^k or $2^{c/2}$, with a security margin of 5 rounds

(42% of the 12 rounds).

The following table summarizes the security claims for ASCON:

Security Aspect	ASCON-128	ASCON-128a	ASCON-180pq
Confidentiality of Plaintext	128 bits	128 bits	128 bits
Integrity of Plaintext	128 bits	128 bits	128 bits
Integrity of Associated Data	128 bits	128 bits	128 bits
Integrity of Public Message Number	128 bits	128 bits	128 bits
Key Recovery Resistance	2^{128}	2^{96}	2^{64}
Collision Resistance (Hashing)	128 bits	128 bits	128 bits

3 | IoT-Applicable Generalized Frameproof Combinatorial Designs

3.1 Introduction

Secret sharing schemes are widely used to protect data by breaking the secret into pieces and sharing them amongst various members of a party. In this chapter, our objective is to produce a repairable ramp scheme that allows for the retrieval of a share through a collection of members in the event of its loss. Repairable Threshold Schemes (RTSs) can be used in cloud storage and General Data Protection Regulation (GDPR) protocols. Secure and energy-efficient data transfer in sensor-based IoTs is built using ramp-type schemes. Protecting personal privacy and reinforcing the security of electronic identification (eID) (Park & Lee, 2018) cards can be achieved using similar schemes. In this chapter, our objective is to produce a repairable ramp scheme that allows for the retrieval of a share through a collection of members in the event of its loss. We propose a combinatorial design that extends the RTS proposed by Kacsmar and Stinson in 2019 (Kacsmar & Stinson, 2019) over the integer ring \mathbb{Z} . Desmedt et al. introduced the concept of frameproofness in 2021 (Desmedt et al., 2021), which motivated us to further improve our construction with respect to this framework. We introduce a graph theoretic approach to the design for a well-rounded and easy presentation of the idea and clarity of our results. We also highlight the importance of secret sharing schemes for IoT applications, as they distribute the secret amongst several devices. Secret sharing schemes offer superior security in lightweight IoT compared to symmetric key encryption or AE schemes because they do not disclose the entire secret to a single device, but rather distribute it among several devices.

The Internet of Things (IoT) is a rapidly expanding network of interconnected devices that communicate with each other to carry out various tasks. With the increasing number of IoT devices comes a growing need for secure communication among them. Cryptography plays a crucial role in ensuring the security of IoT devices, with secret sharing schemes standing out as a promising cryptographic element for IoT applications. One example is the Datachest application (Čuřík, Ploszek, & Zajac, 2022), which encrypts and stores sensitive data in commercial cloud storage systems using secret sharing methods. In this setup, data is uploaded in encrypted form, and the cryptographic keys are divided into shares, with each cloud server receiving

one share. This approach significantly enhances the security of user data stored in the cloud. This chapter highlights the significance of implementing secret sharing schemes in IoT and explores the potential benefits of our proposed distribution design in terms of frameproofness and integrating multiple systems without losing their distinctive characteristics.

Secret sharing schemes are particularly suitable for IoT applications where multiple devices collaborate to perform tasks, such as in a smart home system involving sensors, cameras, and smart locks. In a secret sharing-based IoT (SBIoT) setup, each cloud server receives a share generated using a secret sharing scheme. The energy efficiency of an IoT network, affects its lifespan (Shivhare et al., 2022) and is therefore a critical factor to consider. The use of a ramp-type scheme is known to enhance security and energy efficiency in SBIoT networks (Tang, 2021), ultimately protecting against various types of attacks and data leakages. Implementing a threshold scheme in eID cards (Park & Lee, 2018) can enhance personal information protection by distributing sensitive data between the client and the card, ensuring that even if the card is lost, the original information remains secure due to the distributed storage mechanism. This strategy prevents unauthorized access to personal information by requiring multiple secret shares to reconstruct the original data.

Consider b players and a positive integer $\tau \leq b$. Suppose a dealer distributes a secret to these b players such that any collection of τ players can reconstruct the secret with their shares, but no smaller collection of players can do so. This is called a (τ, b) -threshold secret sharing scheme with *threshold* τ . If the dealer distributes shares to b players such that any collection of τ_1 players can reconstruct the secret but no collection of τ_2 or less players can do so (for $\tau_2 < \tau_1 \leq b$), then it is called a (τ_1, τ_2, b) -ramp scheme. Thus, if $\tau_1 - \tau_2 = 1$, then it is a (τ_1, b) -threshold scheme. In this chapter, we shall present a repairable ramp scheme, which we call a *tensor design*.

A threshold scheme that is secure against all adversaries, irrespective of computational power, is called an *unconditionally secure* threshold scheme. The *information rate* ρ of a threshold scheme is the ratio of the size of the secret to the size of a player's share, i.e.

$$\rho = \frac{\log_2 |\mathcal{K}|}{\log_2 |\mathcal{S}|},$$

where \mathcal{S} is the set of all possible shares and \mathcal{K} is the set of all possible secrets. An *ideal* secret sharing scheme has information rate 1.

In their 2019 work, Stinson and Kacsmar ([Kacsmar & Stinson, 2019](#)) demonstrated non-ideal secret sharing schemes stemming from an ideal scheme (viz. Shamir scheme) as the base scheme. They presented a distribution design which was a threshold scheme with the ability to repair lost shares with a certain probability, and secure against any adversary with fewer players than the threshold. Our work further generalizes the domain over which our distribution designs are defined, in addition to providing it with easier secret reconstruction and share reparability, and securing it in more than one context. In short, we revisit the combinatorial design and some of its key properties first.

3.1.1 Combinatorial RTS

Consider the problem of securely reconstructing the lost share of a player by that player and a subset of the other players. A combinatorial solution to this problem was proposed by Stinson and Wei ([Stinson & Wei, 2018](#)). These schemes are termed *combinatorial RTS*. A *repairable threshold scheme (RTS)* is a (τ, b) -threshold scheme in which a subset of players can repair another player's share in the event that their share is lost or corrupted, without the participation of the dealer who set up the scheme. The repairing protocol should not compromise the (unconditional) security of the threshold scheme.

3.1.2 A Drawback and An Idea of Extension

The combinatorial model proposed so far produces shares that are in a finite field \mathbb{F}_{q^k} . Whether we can extend this notion to an integer ring is the first question. In this work, we propose a method to construct a distribution design with entries from an integer ring, thus generalizing the domain. We further show that this is a ramp scheme and consequently give a method of secret reconstruction for it, which is significantly easier in comparison to ([Kacsmar & Stinson, 2019](#)). The size of the authorized coalition that can recover the secret is significantly reduced in our framework. Example 3 will demonstrate the fact.

Repairability Problem

Techniques from network reliability theory are heavily used in reliability studies of these combinatorial repairable threshold schemes in a setting where players may not be available to take part in the repair of a given player's share. Reference ([Kacsmar & Stinson, 2019](#)) deals

with the problem of reliability of such schemes and reconstruction of secrets and repairing shares without participation of the dealer.

The scheme proposed in this chapter produces a far more efficient share repairability, which is possible due to the generalized domain, and based heavily on the easier secret reconstruction mentioned beforehand.

3.1.3 Frameproofness

Moving forward with the concept of repairing shares, another similar possibility was recently explored, called framing. Instead of simply specifying the minimum size of a set of players that can access the secret, suppose the dealer defines the share distribution through some other process. Say $f : \mathbf{P} \rightarrow \{0, 1\}$ (where \mathbf{P} denotes the power set of the set of all players \mathcal{P}) such that any *coalition* of players $\mathcal{A} \subseteq \mathcal{P}$ can access the secret if and only if $f(\mathcal{A}) = 1$ (thus, in a Shamir scheme, $f(\mathcal{A}) = 1$ if and only if $|\mathcal{A}| \geq \tau$). If $\mathcal{A} \subseteq \mathcal{P}$ maps to 1 through f , then \mathcal{A} is called an *authorized coalition*; if it maps to 0, then \mathcal{A} is an *unauthorized coalition*.

Given such an *access structure* over a secret sharing scheme, suppose a coalition \mathcal{A} of players can gain information about the share of a player $P \in \mathcal{P} \setminus \mathcal{A}$ dishonestly. Then \mathcal{A} can wrongly accuse P of releasing information about the secret that only \mathcal{A} is not authorized to access, i.e., \mathcal{A} can *frame* P . Framing a player (or players) evidently undermines the security of any secret sharing scheme, as it allows a group of players to access extra information about the secret illegally. Thus, it is imperative to limit such capabilities and/or size of any such coalition when constructing a combinatorial RTS. The concept of frameproofness was examined by Desmedt et al. in their recent paper (Desmedt et al., 2021). In this chapter, we improve the extension scheme so that no framing is possible for any coalition of smaller size than the threshold. The question of what can be the minimum size of a coalition that can frame a player under this modification currently remains open.

3.2 Results

In this chapter, we first introduce an operation, the Krönecker product of two matrices, extendable to a Krönecker product of two BIBDs. Following up with some properties of this operation, we present methods to solve two inherent problems with Krönecker products; firstly, the operation

does not produce a BIBD from two BIBDs, and secondly, we resolve the issue of uniqueness that arises with the introduction of this operation. Our next theorem deals with the existence of secret reconstruction, which we prove by producing an algorithm. A probabilistic proof is given next.

An immediate consequence of our results on the new scheme is its extensibility to multiple BIBDs. We discuss it briefly through a dealer's algorithm. We proceed with an example to illustrate our algorithms further. We make considerable improvements on the method of share repair described in (Kacsmar & Stinson, 2019) for our proposed Krönecker product-induced BIBDs.

Next, we explore the concept of frameproofness for our proposed model and improve it significantly through certain changes in the model. We also prove existence of frameproofness of the modified scheme through results based on matchings of bipartite graphs.

Finally, we note the importance of secret sharing schemes in varied IoT applications, especially for their lightweight functionality, uniquely encapsulated through the non-accessibility of the full secret to any single entity, which we strengthen by frameproofness and can expand by incorporating multiple systems by our Krönecker product.

Our chapter starts with a brief review of the work performed by Stinson and Wei (Stinson & Wei, 2018) in Section 3.3. We then move on to describe our construction, beginning with an introduction of the Krönecker product of two BIBDs in Section 3.4. We describe the secret reconstruction procedure for such an object illustrated through an example in Section 3.5. Next, we briefly describe the method of share repair and compute the corresponding repair probabilities, much like in (Kacsmar & Stinson, 2019), in Section 3.6. We then proceed to modify this scheme to give a frameproof construction in Section 3.7. Furthermore, we answer the question of existence of such a modified construction in Section 3.8.

3.3 Stinson and Wei's Model

The classical Shamir scheme is defined over a finite field \mathbb{F}_q ($q \geq b + 1$). It involves the following:

- an *initialization phase*, in which the dealer chooses distinct, non-zero public elements

x_1, x_2, \dots, x_b from \mathbb{F}_q , and gives value x_i to player P_i ;

- a *share distribution phase* in which the dealer chooses a secret $K = a_0 \in \mathbb{F}_q$, then secretly chooses $a_1, \dots, a_{\tau-1} \in \mathbb{F}_q$ independently and uniformly at random, and finally computes the share $y_i = a(x_i)$ (where $a(x) := \sum_{j=0}^{\tau-1} a_j x^j$) and gives it to player P_i .

The combinatorial solution proposed by Stinson and Wei (Stinson & Wei, 2018) to the share repairability problem is based on an old technique by Benaloh and Leichter, namely, giving each player a subset of shares from an underlying threshold scheme called a base scheme (which is, say, a (σ, m) -Shamir scheme over the base field \mathbb{F}_q , where a minimum of σ players out of a total m players can reconstruct the secret). Each player is then given a certain subset of d of the m shares, by use of a set system (or *design*) consisting of b blocks of size d , defined on a set of m points. This design is termed the *distribution design* \mathcal{B} :

$$\begin{pmatrix} y_{11} & y_{12} & \cdots & y_{1d} \\ y_{21} & y_{22} & \cdots & y_{2d} \\ \vdots & & & \\ y_{b1} & y_{b2} & \cdots & y_{bd} \end{pmatrix}, \quad \left| \left\{ y_{ij} \mid \substack{i \in \{1, 2, \dots, b\} \\ j \in \{1, 2, \dots, d\}} \right\} \right| \leq m. \quad (3.1)$$

The resulting expanded (τ, b) -threshold scheme consists of each player P_i corresponding to a block $B_i \in \mathcal{B}$ of the distribution design. For each point $x \in B_i$, the player P_i is given the subshare s_x . If X denotes the set of m points on which the design is defined and $\mathcal{B} = \{B_1, \dots, B_b\}$ is the set of all blocks, then this forms an (X, \mathcal{B}) -distribution design.

Recall Definition 2.1 of a BIBD. Observe that if each point occurs in exactly r blocks, then the parameters b, v, k, r, λ of a BIBD satisfy the following relations (Stinson, 2004):

- (i) $bk = vr$;
- (ii) $\lambda(v - 1) = r(k - 1)$;
- (iii) $b \geq v$ (and hence $r > k$).

Definition 3.1 We shall call a distribution design a tensor design if it simply satisfies property (i) above.

Design Properties

For the purpose of computations, we recall some results from (Kacsmar & Stinson, 2019) on block designs.

Theorem 3.1 (Replication Number) *Every point in a (v, k, λ) -BIBD occurs in exactly $r = \frac{\lambda(v-1)}{k-1}$ blocks. The value r is termed the replication number of the scheme.*

Theorem 3.2 (Blocks and Block Size) *A (v, k, λ) -BIBD has exactly $b = \frac{vr}{k} = \frac{\lambda(v^2-v)}{k^2-k}$ blocks of size k .*

3.4 Tensor Design Generated by Two BIBDs

Given two matrices \mathcal{A} and \mathcal{B} , the usual matrix product operation can be carried out only when the column size of the left matrix \mathcal{A} is equal to the row size of the right matrix \mathcal{B} . The Krönecker product can be applied on any two matrices, irrespective of their dimension. This operation has several applications in Linear Algebra, of which, we consider some properties that shall be useful for working with BIBDs.

3.4.1 Definition of the Krönecker Product

The *Krönecker product* of two matrices $\mathcal{A}_{b_1 \times k_1}$ and $\mathcal{B}_{b_2 \times k_2}$ is the block matrix

$$\mathcal{A} \otimes \mathcal{B} = \begin{pmatrix} \mathbf{a}_{11}\mathcal{B} & \mathbf{a}_{12}\mathcal{B} & \dots & \mathbf{a}_{1k_1}\mathcal{B} \\ \mathbf{a}_{21}\mathcal{B} & \mathbf{a}_{22}\mathcal{B} & \dots & \mathbf{a}_{2k_1}\mathcal{B} \\ \vdots & & & \\ \mathbf{a}_{b_11}\mathcal{B} & \mathbf{a}_{b_12}\mathcal{B} & \dots & \mathbf{a}_{b_1k_1}\mathcal{B} \end{pmatrix}, \quad (3.2)$$

where \mathbf{a}_{ij} denotes the entry in the i^{th} row and j^{th} column of \mathcal{A} .

Observe that Krönecker products follow the associative property. Thus, for matrices \mathcal{A} , \mathcal{B} , and \mathcal{C} ,

$$(\mathcal{A} \otimes \mathcal{B}) \otimes \mathcal{C} = \mathcal{A} \otimes (\mathcal{B} \otimes \mathcal{C}).$$

Another interesting property of Krönecker products is that they maintain structure over block

matrices. Thus, if \mathcal{A} is written as a block matrix

$$\begin{pmatrix} \mathcal{A}_{11} & \mathcal{A}_{12} & \cdots & \mathcal{A}_{1k} \\ \mathcal{A}_{21} & \mathcal{A}_{22} & \cdots & \mathcal{A}_{2k} \\ \vdots & & & \\ \mathcal{A}_{b1} & \mathcal{A}_{b2} & \cdots & \mathcal{A}_{bk} \end{pmatrix} \text{ for some } b \leq b_1 \text{ and } k \leq k_1,$$

$$\text{then } \mathcal{A} \otimes \mathcal{B} = \begin{pmatrix} \mathcal{A}_{11} \otimes \mathcal{B} & \mathcal{A}_{12} \otimes \mathcal{B} & \cdots & \mathcal{A}_{1k} \otimes \mathcal{B} \\ \mathcal{A}_{21} \otimes \mathcal{B} & \mathcal{A}_{22} \otimes \mathcal{B} & \cdots & \mathcal{A}_{2k} \otimes \mathcal{B} \\ \vdots & & & \\ \mathcal{A}_{b1} \otimes \mathcal{B} & \mathcal{A}_{b2} \otimes \mathcal{B} & \cdots & \mathcal{A}_{bk} \otimes \mathcal{B} \end{pmatrix}. \quad (3.3)$$

3.4.2 Krönecker Product of Two BIBDs

Let \mathcal{A} and \mathcal{B} be the share matrices generated by ramp schemes with, respectively, b_1 and b_2 blocks having shares of sizes k_1 and k_2 . Suppose \mathcal{A} and \mathcal{B} also denote the $b_1 \times k_1$ and $b_2 \times k_2$ matrices corresponding to the two schemes. The Krönecker product of $\mathcal{A} \otimes \mathcal{B}$ is therefore

$$M = \begin{pmatrix} \mathbf{a}_{11}\mathcal{B} & \mathbf{a}_{12}\mathcal{B} & \cdots & \mathbf{a}_{1k_1}\mathcal{B} \\ \mathbf{a}_{21}\mathcal{B} & \mathbf{a}_{22}\mathcal{B} & \cdots & \mathbf{a}_{2k_1}\mathcal{B} \\ \vdots & & & \\ \mathbf{a}_{b_11}\mathcal{B} & \mathbf{a}_{b_12}\mathcal{B} & \cdots & \mathbf{a}_{b_1k_1}\mathcal{B} \end{pmatrix} = \begin{pmatrix} T_1 \\ T_2 \\ \vdots \\ T_{b_1} \end{pmatrix}, \quad (3.4)$$

where T_i ($i \in \{1, 2, \dots, b_1\}$) is the i^{th} row-block submatrix of M containing rows $(i-1)b_2 + 1, (i-1)b_2 + 2, \dots, ib_2$. If the share matrix \mathcal{A} is defined over the field \mathbb{F}_{p_1} and \mathcal{B} over the field \mathbb{F}_{p_2} for some primes p_1 and p_2 , then we define the scalar multiplication by simple integer multiplication:

$$\mathbb{F}_{p_1} \times \mathbb{F}_{p_2} \rightarrow \mathbb{Z}$$

$$\text{such that } (x_1, x_2) \mapsto x_1 \cdot x_2.$$

The reason behind taking such a multiplication is that the product elements are not distinguishable from integers. Therefore, M is a matrix over the integer ring \mathbb{Z} .

At this point, the first observation that we make is that the Krönecker product $\mathcal{A} \otimes \mathcal{B}$ of two BIBDs \mathcal{A} and \mathcal{B} does not always produce a BIBD. To illustrate the fact, we start with a small example, and then we describe a method for resolving this issue. Also, the Krönecker product in general does not produce an injective mapping from $\mathcal{M}_{b_1 \times k_1} \times \mathcal{M}_{b_2 \times k_2}$ to the matrix space $\mathcal{M}_{b_1 b_2 \times k_1 k_2}$. So it is hopeless to search for a secret reconstruction procedure from a given Krönecker product matrix. We shall thus impose a condition producing an injective map and in turn, ensuring the existence of secret reconstruction.

Consider an example of two $(4, 3, 2)$ Shamir schemes in \mathbb{F}_5 and \mathbb{F}_7 over the points $\{1, 2, 3, 4\}$ and $\{1, 2, 3, 5\}$ constructed using two polynomials modulo \mathbb{F}_5 and \mathbb{F}_7 , respectively. These can be represented by share matrices \mathcal{A} and \mathcal{B} , respectively, with $r_1 = r_2 = 3$:

$$\mathcal{A} = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 4 \\ 3 & 4 & 2 \\ 4 & 3 & 1 \end{pmatrix} \text{ and } \mathcal{B} = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 5 \\ 3 & 5 & 1 \\ 5 & 1 & 2 \end{pmatrix}. \quad (3.5)$$

The Krönecker product of the BIBDs \mathcal{A} and \mathcal{B} is as follows:

1	2	3	2	4	6	3	6	9
2	3	5	4	6	10	6	9	15
3	5	1	6	10	2	9	15	3
5	1	2	10	2	4	15	3	6
2	4	6	1	2	3	4	8	12
4	6	10	2	3	5	8	12	20
6	10	2	3	5	1	12	20	4
10	2	4	5	1	2	20	4	8
3	6	9	4	8	12	2	4	6
6	9	15	8	12	20	4	6	10
9	15	3	12	20	4	6	10	2
15	3	6	20	4	8	10	2	4
4	8	12	3	6	9	1	2	3
8	12	20	6	9	15	2	3	5
12	20	4	9	15	3	3	5	1
20	4	8	15	3	6	5	1	2

Hence, $\mathcal{A} \otimes \mathcal{B}$ has the parameters $b = 16, v = 12$, and $k = 9$; the parameters r and λ are not well-defined. Obviously, neither does this satisfy property 3 of a BIBD (Definition 2.1), nor the relation (i) of a tensor design (Definition 3.1). Lemmas 3.1, 3.2, 3.3 and Theorem 3.3 ensure that we always obtain a tensor design from a Krönecker product, and furthermore that we always obtain a secret reconstruction for such a share distribution scheme.

3.4.3 Some Results on the Krönecker Product of BIBDs

We now resolve these issues by defining some properties of a tensor design. Let \mathcal{A} and \mathcal{B} be share matrices defined on points $\{x_1, x_2, \dots, x_n\}$ and $\{y_1, y_2, \dots, y_m\}$, respectively. Let \mathcal{B}_d be the same distribution scheme as \mathcal{B} , but on the points $\{y_1 + d, y_2 + d, \dots, y_m + d\}$. The position of an element in the Krönecker product of these two matrices can be found by simple counting, and is stated in the following lemma:

Lemma 3.1 *The product of $a_{ij} \in \mathcal{A}$ and $b_{kl} \in \mathcal{B}$ can be found in the row $(i - 1)b_2 + k$ (which is also the player number in the repair scheme represented by M), and the column $(j - 1)k_2 + l$ of M .*

The next result helps ensure that $\mathcal{A} \otimes \mathcal{B}$ is indeed a BIBD:

Lemma 3.2 *Let $\{x_1, x_2, \dots, x_n\}$ and $\{y_1, y_2, \dots, y_m\}$ be two collections of integers. Then there exists an integer d such that $\{x_1, x_2, \dots, x_n\}$ and $\{y_1 + d, y_2 + d, \dots, y_m + d\}$ have no multiplicative collisions of the type $x_i y_j = x_k y_l$ for $(i, j) \neq (k, l)$.*

Proof: Set $d \geq \max_{\substack{i,k \in \{1,2,\dots,n\} \\ j,l \in \{1,2,\dots,m\}}} \{x_i y_j - x_k y_l\} + 1$. Suppose $x_i(y_j + d) = x_k(y_l + d)$.

$$\begin{aligned} \implies x_i y_j + x_i d &= x_k y_l + x_k d \\ \implies (x_k - x_i) d &= x_i y_j - x_k y_l \\ \implies d &= \frac{x_i y_j - x_k y_l}{x_k - x_i}; \end{aligned} \tag{3.6}$$

however, since $d \geq \max_{\substack{i,k \in \{1,2,\dots,n\} \\ j,l \in \{1,2,\dots,m\}}} \{x_i y_j - x_k y_l\} + 1$, this is a contradiction. Therefore, $\{x_1, x_2, \dots, x_n\}$ and $\{y_1 + d, y_2 + d, \dots, y_m + d\}$ produce no multiplicative collisions. \square

Lemma 3.3 *Given a list of distinct elements $\{y_1, y_2, \dots, y_m\}$, we can choose an integer \hat{d} such that $\gcd(y_1 + \hat{d}, y_2 + \hat{d}, \dots, y_m + \hat{d}) = 1$.*

Proof: Without loss of generality, we may assume $y_1 < y_2 < \dots < y_m$. Let $l = \gcd(y_1, y_2, \dots, y_m)$ and fix $i < j$ in $\{1, 2, \dots, m\}$. Thus, $y_i = lk_i$ and $y_j = lk_j$ such that $k_i < k_j$. Choose \hat{d} such that $\gcd(\hat{d}, l) = 1$ and $\gcd(\hat{d} + y_i, k_j - k_i) = 1$ for some j in $\{1, 2, \dots, m\}$. Now, $\gcd(y_i + \hat{d}, y_j + \hat{d}) = \gcd(lk_i + \hat{d}, lk_j + \hat{d}) = \gcd(lk_i + \hat{d}, l(k_j - k_i)) = 1$. \square

Theorem 3.3 (Reconstruction from Tensor Designs) *Consider a*

$(v_1, k_1, \lambda_1, b_1, r_1)$ -BIBD \mathcal{A} and a $(v_2, k_2, \lambda_2, b_2, r_2)$ -BIBD \mathcal{B} . Also consider a (public) integer d such that there are no multiplicative collisions of the type $x_i(y_j + d) = x_k(y_l + d)$ for $(i, j) \neq (k, l)$.

1. *The matrix $\mathcal{A} \otimes \mathcal{B}_d$ produces a tensor design (over the integer ring \mathbb{Z}).*
2.
 - *If $\gcd(x_1, x_2, \dots, x_{v_1}) = 1$, and*
 - *$\gcd(y_1, y_2, \dots, y_{v_2}) = 1$,*

then \mathcal{A} and \mathcal{B} can be reproduced from a collection of players in the new scheme $\mathcal{A} \otimes \mathcal{B}_d$, hence enabling share repair and secret reconstruction.

This theorem can be generalized for finitely many such Krönercker products, and motivates us to present the following algorithm for a share distribution scheme.

Proof: The parameters of the Krönercker product $\mathcal{A} \otimes \mathcal{B}$ are $b = b_1 b_2, v = v_1 v_2, k = k_1 k_2, r = r + 1r + 2, \lambda = \lambda_1 \lambda_2$. Part 1 of the theorem therefore follows from Lemma 3.2, which ensures a well-defined value for r , and Lemma 3.3, which ensures a well-defined value for λ .

In order to prove part 2, we describe two ways to reproduce \mathcal{A} and \mathcal{B} . Recall first that any τ_1 rows of \mathcal{A} produce all points of \mathcal{A} , and similarly τ_2 rows for \mathcal{B}_d . Furthermore, we claim the following:

[I] A collection of players that has

- (i) τ_2 players from one row-block T_i of M ;
- (ii) at least one player from distinct $\tau_1 - 1$ row-blocks $T_j \neq T_i$ of the remaining $b_1 - 1$ row-blocks

can reconstruct the secret.

[II] Let S_j ($j \in \{1, 2, \dots, b_2\}$) be the collection of players $\{P_{b_2 k + j} : k \in \{0, 1, \dots, b_1 - 1\}\}$. A collection of players that contains

- (i) τ_1 players from one S_j ;
- (ii) at least one player from $\tau_2 - 1$ $S_i, i \neq j$

can also reconstruct the secret.

We now present an algorithm to prove claim [I]; claim [II] follows similarly.

1. The share of the j^{th} player $P_{i \cdot b_2 - 1 + j}$ of the i^{th} row-block T_i is of the form

$$\mathbf{a}_{i1} \cdot \{\mathbf{b}_{j_1}, \mathbf{b}_{j_2}, \dots, \mathbf{b}_{j_{k_2}}\}, \mathbf{a}_{i2} \cdot \{\mathbf{b}_{j_1}, \mathbf{b}_{j_2}, \dots, \mathbf{b}_{j_{k_2}}\}, \dots, \mathbf{a}_{ik_1} \cdot \{\mathbf{b}_{j_1}, \mathbf{b}_{j_2}, \dots, \mathbf{b}_{j_{k_2}}\}.$$

Fix any $i \in \{1, 2, \dots, b_1\}$ and choose $j_1, j_2, \dots, j_{\tau_2}$ to ensure that

$$\gcd(\mathbf{b}_{j_1 1}, \mathbf{b}_{j_1 2}, \dots, \mathbf{b}_{j_1 k_2}, \mathbf{b}_{j_2 1}, \mathbf{b}_{j_2 2}, \dots, \mathbf{b}_{j_2 k_2}, \dots, \mathbf{b}_{j_{\tau_2} 1}, \mathbf{b}_{j_{\tau_2} 2}, \dots, \mathbf{b}_{j_{\tau_2} k_2}) = 1.$$

2. Therefore, the values of $\mathbf{a}_{i1}, \mathbf{a}_{i2}, \dots, \mathbf{a}_{ik_1}$ become known. Divide $\mathbf{a}_{i\alpha} \mathbf{b}_{j_k \beta}$ by $\mathbf{a}_{i\alpha}$ (for $\alpha \in \{1, 2, \dots, k_1\}, \beta \in \{1, 2, \dots, k_2\}$ and $k \in \{1, 2, \dots, \tau_2\}$) to obtain $\mathbf{b}_{j_k 1}, \mathbf{b}_{j_k 2}, \dots, \mathbf{b}_{j_k k_2}$.

3. Construct the complete matrix \mathcal{B}_d using the shares of τ_2 players of \mathcal{B}_d that are now known. Hence construct \mathcal{B} .
4. Using the values of the elements in \mathcal{B}_d , compute the values $\mathbf{a}_{i'1}, \mathbf{a}_{i'2}, \dots, \mathbf{a}_{i'k_1}$ for $\tau_1 - 1$ indices i' that are distinct from each other as well as from i .
5. Hence, construct \mathcal{A} from the shares of τ_1 players of \mathcal{A} thus obtained.
6. Finally compute the secret from \mathcal{A} and \mathcal{B} .

□

This reconstruction algorithm is clearly better than the one in (Kacsmar & Stinson, 2019) in the sense that the size of the authorized coalition is smaller. In fact, the size of the authorized coalition, while not unique, has a lower bound in the number of players. The following section provides a proof that there is always a secret reconstruction for this scheme.

3.4.4 Proof of Existence of Secret Reconstruction

Let us redefine the problem in terms of random variables. Let X_1, X_2, \dots, X_n be sampled without replacement from the collection of all players.

$$\text{Let } I_{i,j} = \begin{cases} 1 & \text{if } X_i \in S_j, i \in [n], j \in [b_2], \\ 0 & \text{otherwise.} \end{cases}$$

$$\text{Also let } J_{i,k} = \begin{cases} 1 & \text{if } X_i \in T_k, i \in [n], k \in [b_1], \\ 0 & \text{otherwise.} \end{cases}$$

We further define $n_k = \sum_{i=1}^n J_{i,k}$ and $r_j = \sum_{i=1}^n I_{i,j}$. Then the condition for reconstruction becomes

[I] (i) $\max_{k \in [b_1]} n_k \geq \tau_2$,

(ii) $n_k \geq 1$ for at least τ_1 indices k .

[II] (i) $\max_{j \in [b_2]} r_j \geq \tau_1$,

(ii) $r_j \geq 1$ for at least τ_2 indices j .

Let E_1 be the event that condition [I] is satisfied and E_2 be the event that condition [II] is satisfied. Also, let $D(n_0)$ be the event that $n \geq n_0$. We find an n_0 such that $\Pr [E_1 \cup E_2 | n \geq n_0] \approx 1$. This is equivalent to $\Pr [E_1^c \cap E_2^c | n \geq n_0] \approx 0$. In fact, it is sufficient to show $\Pr [E_1^c | n \geq n_0] \approx 0$ and $\Pr [E_2^c | n \geq n_0] \approx 0$.

As $E_1 = E_1(i) \cap E_1(ii)$, $E_1^c = E_1(i)^c \cup E_1(ii)^c$,

$$\begin{aligned} \Pr [E_1^c | n \geq n_0] &= \Pr [E_1(i)^c \cup E_1(ii)^c | n \geq n_0] \\ &= \Pr [E_1(i)^c | n \geq n_0] + \Pr [E_1(ii)^c | n \geq n_0] - \Pr [E_1(i)^c \cap E_1(ii)^c | n \geq n_0] \end{aligned}$$

Lemma 3.4 $\Pr [E_1(i)^c \cap E_1(ii)^c | n \geq (\tau_1 - 1)(\tau_2 - 1) + 1] = 0$.

Proof: We observe that $E_1(i)^c$ is the event $\max_{k \in [b_1]} n_k < \tau_2$ and $E_1(ii)^c$ is the event that $n_k \geq 1$ for at most $\tau_1 - 1$ indices k . Thus, if there are $(\tau_1 - 1)(\tau_2 - 1) + 1$ players in a collection, then by the pigeonhole principle, either $E_1(i)^c$ or $E_1(ii)^c$ is violated. \square

Lemma 3.5 $\Pr [E_1(i)^c | n \geq (\tau_2 - 1)b_1 + 1] = 0$.

Proof: We observe that $E_1(i)^c$ is the event $\max_{k \in [b_1]} n_k < \tau_2$ and there are b_1 n_k s. Thus, if there are $(\tau_2 - 1)b_1 + 1$ players in a collection, then by the pigeonhole principle, $E_1(i)^c$ is violated, since there is at least one n_k with τ_2 or more players. \square

Lemma 3.6 $\Pr [E_1(ii)^c | n \geq (\tau_1 - 1)b_2 + 1] = 0$.

Proof: We observe that $E_1(ii)^c$ is the event that $n_k \geq 1$ for at most $\tau_1 - 1$ indices k . By definition, each n_k can have at most b_2 elements. Thus, any collection of $(\tau_1 - 1)b_2 + 1$ players violates $E_1(ii)^c$. \square

Lemma 3.7 $\Pr [E_2(i)^c \cap E_2(ii)^c | n \geq (\tau_1 - 1)(\tau_2 - 1) + 1] = 0$.

Proof: We observe that $E_2(i)^c$ is the event $\max_{j \in [b_2]} r_j < \tau_1$ and $E_2(ii)^c$ is the event that $r_j \geq 1$ for at most $\tau_2 - 1$ indices j . Thus, if there are $(\tau_1 - 1)(\tau_2 - 1) + 1$ players in a collection, then by the pigeonhole principle, either $E_2(i)^c$ or $E_2(ii)^c$ is violated. \square

Lemma 3.8 $\Pr [E_2(i)^c | n \geq (\tau_1 - 1)b_2 + 1] = 0$.

Proof: We observe that $E_2(i)^c$ is the event $\max_{j \in [b_2]} r_j < \tau_1$ and there are b_2 r_j s. Thus, if there are $(\tau_1 - 1)b_2 + 1$ players in a collection, then by the pigeonhole principle, $E_2(i)^c$ is violated, since there is at least one r_j with τ_1 or more players. \square

Lemma 3.9 $\Pr [E_2(ii)^c \mid n \geq (\tau_2 - 1)b_1 + 1] = 0$.

Proof: We observe that $E_2(ii)^c$ is the event that $r_j \geq 1$ for at most $\tau_2 - 1$ indices j . By definition, each r_j can have at most b_1 elements. Thus, any collection of $(\tau_2 - 1)b_1 + 1$ players violates $E_2(ii)^c$. \square

For $n_0 = \max\{(\tau_2 - 1)b_1 + 1, (\tau_1 - 1)b_2 + 1\}$, Lemmas 3.4, 3.5, and 3.6 imply $\Pr [E_1^c \mid n \geq n_0] = 0$ and $n_0 = \max\{(\tau_2 - 1)b_1 + 1, (\tau_1 - 1)b_2 + 1\}$, and Lemmas 3.7, 3.8, and 3.9 imply $\Pr [E_2^c \mid n \geq n_0] = 0$.

Note that the bound given here for the reconstruction number *is tight*, as we might expect. In the example presented in Section 3.5, the bound turns out to be 5, which matches all the bounds above. Corresponding counterexamples can be constructed to show that no smaller-sized general collection can complete the reconstruction.

This result can be generalized for three or more designs. These results provide us with the tools to present a generalized scheme, which we do now.

3.4.5 A Generalized Share Distribution Scheme

1. Dealer selects n (not necessarily distinct) BIBDs $\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_n$, where for $i \in \{1, 2, \dots, n\}$, \mathcal{A}_i is defined over points $\{x_1^i, x_2^i, \dots, x_{v_i}^i\}$.
2. Dealer finds an integer d_1 such that $\gcd(x_1^1 + d_1, x_2^1 + d_1, \dots, x_{v_1}^1 + d_1) = 1$.
3. For $i \in \{2, \dots, n\}$:
 - Dealer finds an integer d_i (using Lemmas 3.2 and 3.3) such that d_i breaks all pairwise multiplicative collisions and makes the gcd of all elements $x_l^j + d_j$ ($j \in \{1, \dots, i-1\}$, $l \in \{1, \dots, v_j\}$) and $x_1^i + d_i, x_2^i + d_i, \dots, x_{v_i}^i + d_i$ is 1.
4. $M \leftarrow \mathcal{A}_1 \otimes \mathcal{A}_2 \otimes \dots \otimes \mathcal{A}_n$.
5. Dealer distributes each row i of M as share to player P_i and outputs (d_1, d_2, \dots, d_n) publicly.

Note that by Theorem 3.3, M is a tensor design, and the algorithm in the proof of the theorem can be generalized for secret reconstruction of this scheme.

3.5 Example

We quickly revisit the previous example demonstrating the Krönecker product of BIBDs \mathcal{A} and \mathcal{B} as in Equation (3.5):

Two $(4, 3, 2)$ Shamir schemes in \mathbb{F}_5 and \mathbb{F}_7 over the points $\{1, 2, 3, 4\}$ and $\{1, 2, 3, 5\}$ can be represented by share matrices \mathcal{A} and \mathcal{B} , respectively, with $r_1 = r_2 = 3$:

$$\mathcal{A} = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 4 \\ 3 & 4 & 2 \\ 4 & 3 & 1 \end{pmatrix} \text{ and } \mathcal{B} = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 5 \\ 3 & 5 & 1 \\ 5 & 1 & 2 \end{pmatrix}.$$

The Krönecker product of the BIBDs \mathcal{A} and \mathcal{B} is as follows:

1	2	3	2	4	6	3	6	9
2	3	5	4	6	10	6	9	15
3	5	1	6	10	2	9	15	3
5	1	2	10	2	4	15	3	6
2	4	6	1	2	3	4	8	12
4	6	10	2	3	5	8	12	20
6	10	2	3	5	1	12	20	4
10	2	4	5	1	2	20	4	8
3	6	9	4	8	12	2	4	6
6	9	15	8	12	20	4	6	10
9	15	3	12	20	4	6	10	2
15	3	6	20	4	8	10	2	4
4	8	12	3	6	9	1	2	3
8	12	20	6	9	15	2	3	5
12	20	4	9	15	3	3	5	1
20	4	8	15	3	6	5	1	2

Using the algorithm in Section 3.4.5, we produce a tensor design $\mathcal{A} \otimes \mathcal{B}_{21}$ using an integer $d = 21$ satisfying Lemma 3.3. Representing the share matrix modified from \mathcal{B} by \mathcal{B}_{21} (and noting that both share matrices are undeclared), with $r_1 = r_2 = 3$:

$$\mathcal{B}_{21} = \begin{pmatrix} 22 & 23 & 24 \\ 23 & 24 & 26 \\ 24 & 26 & 22 \\ 26 & 22 & 23 \end{pmatrix}, \quad (3.7)$$

we still have $b_1 = 4$, $b_2 = 4$, $k_1 = 3$, and $k_2 = 3$. Observe that $\tau_1 = 2$ and $\tau_2 = 2$ are the reconstruction numbers of \mathcal{A} and \mathcal{B} , respectively. The Krönercker product of the two matrices \mathcal{A} and \mathcal{B}_{21} , represented by the matrix M , is shown in Figure 3.1.

22	23	24	44	46	48	66	69	72	$T_1=\{P_1,P_2,P_3,P_4\}$	
23	24	26	46	48	52	69	72	78		
24	26	22	48	52	44	72	78	66		
26	22	23	52	44	46	78	66	69		
44	46	48	22	23	24	88	92	96	$T_2=\{P_5,P_6,P_7,P_8\}$	
46	48	52	23	24	26	92	96	104		
48	52	44	24	26	22	96	104	88		
52	44	46	26	22	23	104	88	92		
66	69	72	88	92	96	44	46	48	$T_3=\{P_9,P_{10},P_{11},P_{12}\}$	
69	72	78	92	96	104	46	48	52		
72	78	66	96	104	88	48	52	44		
78	66	69	104	88	92	52	44	46		
88	92	96	66	69	72	22	23	24	$T_4=\{P_{13},P_{14},P_{15},P_{16}\}$	
92	96	104	69	72	78	23	24	26		
96	104	88	72	78	66	24	26	22		
104	88	92	78	66	69	26	22	23		
$S_1=\{P_1,P_5,P_9,P_{13}\}$			$S_2=\{P_2,P_6,P_{10},P_{14}\}$			$S_3=\{P_3,P_7,P_{11},P_{15}\}$			$S_4=\{P_4,P_8,P_{12},P_{16}\}$	

Figure 3.1: The matrix $\mathcal{A} \otimes \mathcal{B}_{21}$ is the Krönercker product of \mathcal{A} and \mathcal{B}_{21} as in Equation (3.7), and is a secret sharing scheme with reconstruction number 2. A secret reconstruction algorithm for this scheme is detailed in Section 3.5.1.

3.5.1 Secret Reconstruction

The matrix $\mathcal{A} \otimes \mathcal{B}_{21}$ in the above example produces interesting results.

1. A collection of three players—exactly two from one of the sets T_1, T_2, T_3, T_4 and one from another—allows reconstruction of the secret. For example, consider the set of three

players $\{P_1, P_2, P_5\}$. This set can reconstruct the secret:

- (i) $\gcd(22, 23, 24, 23, 24, 26) = 1$; hence, the first row of $M_{\mathcal{A}}$ is $(1 \ 2 \ 3)$ and the first two rows of $M_{\mathcal{B}}$ are $(22 \ 23 \ 24)$ and $(23 \ 24 \ 26)$. As $\tau_2 = 2$, $M_{\mathcal{B}}$ can be obtained from its two rows.
- (ii) Now, observing $5 = 4 \cdot 1 + 1$, we readily know P_5 uses the first row of $M_{\mathcal{B}}$ and the second row of $M_{\mathcal{A}}$; this yields the second row of $M_{\mathcal{A}}$, $(2 \ 1 \ 4)$. Since $\tau_1 = 2$ and we have two rows of $M_{\mathcal{A}}$, the whole matrix $M_{\mathcal{A}}$ is known.

2. Any collection of three players—two from one of the sets S_1, S_2, S_3, S_4 and one from another—also allows reconstruction of the secret.
3. Reconstruction of the secret is ensured for a collection of five or more players.

This idea can be generalized to a secret reconstruction algorithm in the general case.

3.6 Share Repair for a Krönecker Product-Induced Distribution Design

Let \mathcal{A} and \mathcal{B} be $(v_1, k_1, 1)$ - and $(v_2, k_2, 1)$ -BIBDs with b_1 and b_2 blocks, and replication numbers r_1 and r_2 , respectively. Consider player P_1 , whose share is the first block (i.e., row) of $\mathcal{A} \otimes \mathcal{B}$. Thus,

$$\begin{aligned} \text{share of } P_1 = & \mathbf{a}_{11}\mathbf{b}_{11} \ \mathbf{a}_{11}\mathbf{b}_{12} \ \cdots \ \mathbf{a}_{11}\mathbf{b}_{1k_2} \mid \mathbf{a}_{12}\mathbf{b}_{11} \ \mathbf{a}_{12}\mathbf{b}_{12} \ \cdots \ \mathbf{a}_{12}\mathbf{b}_{1k_2} \mid \cdots \\ & \cdots \mid \mathbf{a}_{1k_1}\mathbf{b}_{11} \ \mathbf{a}_{1k_1}\mathbf{b}_{12} \ \cdots \ \mathbf{a}_{1k_1}\mathbf{b}_{1k_2} = L_1 \mid L_2 \mid \cdots \mid L_{k_1}. \end{aligned}$$

Using the notations and method described in (Kacsmar & Stinson, 2019) (and making the same assumption that any player is available with a fixed probability p), the probability of availability of at least one repair set is

$$R(p) = (1 - (1 - p)^{r_1 r_2})^{k_1 k_2}. \quad (3.8)$$

We improve this method significantly. For this, observe that each block L_k

($k \in \{1, 2, \dots, k_1\}$) (possibly with a different factor \mathbf{a}_{mi} for some $m \in \{1, 2, \dots, \dots, b_1\}$, $i \in \{1, 2, \dots, k_1\}$, from \mathcal{A}) occurs in the shares of $r_1 - 1$ players other than P_1 . (3.9)

Furthermore, the share of P_1 can also be characterized as

$$\mathbf{a}_{11} \mathbf{b}_{11} \mathbf{a}_{11} \mathbf{b}_{12} \cdots \mathbf{a}_{11} \mathbf{b}_{1k_2} \mid \mathbf{a}_{12} \mathbf{b}_{11} \cdots \mathbf{a}_{12} \mathbf{b}_{1k_2} \mid \cdots \cdots \mid \mathbf{a}_{1k_1} \mathbf{b}_{11} \cdots \mathbf{a}_{1k_1} \mathbf{b}_{1k_2};$$

$$K_1 := \mathbf{a}_{11} \mathbf{b}_{11} \mathbf{a}_{12} \mathbf{b}_{11} \cdots \mathbf{a}_{1k_1} \mathbf{b}_{11},$$

$$K_2 := \mathbf{a}_{11} \mathbf{b}_{12} \mathbf{a}_{12} \mathbf{b}_{12} \cdots \mathbf{a}_{1k_1} \mathbf{b}_{12},$$

\vdots

$$K_{k_2} := \mathbf{a}_{11} \mathbf{b}_{1k_2} \mathbf{a}_{12} \mathbf{b}_{1k_2} \cdots \mathbf{a}_{1k_1} \mathbf{b}_{1k_2}.$$

It is thus clear that each K_j ($j \in \{1, 2, \dots, k_2\}$) (possibly with a different

factor \mathbf{b}_{lj} for some $l \in \{1, 2, \dots, b_2\}$, from \mathcal{B}) occurs in the shares of $r_2 - 1$

players other than P_1 . (3.10)

Let us assume that we have t_1 players of type (3.9) and t_2 players of type (3.10). Then

$$R_{(t_1, t_2)}^*(p) = R_{t_1}^*(p) R_{t_2}^*(p) R_{\delta}^*(p), \tag{3.11}$$

where

- (i) t_1 are selected from type (3.9);
- (ii) t_2 are selected from type (3.10);
- (iii) $\delta := k_1 k_2 - t_1 k_1 - t_2 (k_2 - t_1)$ are selected independently, and

$$\begin{aligned}
R_{t_1}^*(p) &= (1 - (1 - p)^{r_1 - 1})^{t_1} \\
R_{t_2}^*(p) &= (1 - (1 - p)^{r_2 - 1})^{t_2} \\
R_\delta^*(p) &= (1 - (1 - p)^{(r_1 - 1)(r_2 - 1)})^\delta.
\end{aligned}$$

Observe that $\delta = (k_1 - t_2)(k_2 - t_1)$. Therefore, the probability of at least one repair set being available in this case is

$$R^*(p) = \sum_{t_1, t_2} R_{t_1}^*(p) R_{t_2}^*(p) R_\delta^*(p).$$

Let $E^*(p)$ be the expected number of minimal repair sets. In general, this expected number is the product of the total number of possible repair sets and the probability of availability of each repair set. Ref. (Kacsmar & Stinson, 2019) sets $E(p) = (r_1 r_2)^{k_1 k_2}$. We denote by $C(t_1, t_2)$, the number of partitions of a set of size $k_1 k_2$ into three sets of sizes t_1, t_2 and $k_1 k_2 - t_1 - t_2$. By an argument similar to the previous,

$$\begin{aligned}
E_{t_1}^*(p) &= (r_1 - 1)^{t_1} p^{t_1}, \\
E_{t_2}^*(p) &= (r_2 - 1)^{t_2} p^{t_2}, \text{ and} \\
E_\delta^*(p) &= [(r_1 - 1)(r_2 - 1)]^\delta p^\delta, \text{ so that} \\
E_{(t_1, t_2)}^*(p) &= C(t_1, t_2) E_{t_1}^*(p) E_{t_2}^*(p) E_\delta^*(p). \\
\text{Hence, } E^*(p) &= \sum_{t_1, t_2} C(t_1, t_2) E_{t_1}^*(p) E_{t_2}^*(p) E_\delta^*(p).
\end{aligned}$$

Table 3.1 shows a comparison of share repair probability on three projective planes for two different methods.

Table 3.1: A comparison table showing probability of share repairability on three projective planes.

\mathcal{A}	\mathcal{B}	$R(p)$	$R^*(p)$
(3, 2, 1)	(3, 2, 1)	$(1 - q^3)^4$	$> (1 - q)^4 + \dots$
(3, 2, 1)	(7, 3, 1)	$(1 - q^5)^6$	$> (1 - q^2)^6 + \dots$
(7, 3, 1)	(7, 3, 1)	$(1 - q^8)^9$	$> (1 - q^4)^9 + \dots$

3.7 Frameproofness

Consider matrix representations of two BIBDs $\mathcal{A} = (\mathbf{a}_{ij})_{\substack{i \in \{1, \dots, b_1\} \\ j \in \{1, \dots, k_1\}}}$ and $\mathcal{B} = (\mathbf{b}_{ij})_{\substack{i \in \{1, \dots, b_2\} \\ j \in \{1, \dots, k_2\}}}$, and their Krönecker product as depicted in Equation (3.4). We show here how the share of a player, say P_1 , can be retrieved (i.e., player P_1 can be *framed*; see (Desmedt et al., 2021) for more details) by only two other players. For clarity, we mention here that the share of P_1 is

$$\mathbf{a}_{11}\mathbf{b}_{11}, \mathbf{a}_{11}\mathbf{b}_{12}, \dots, \mathbf{a}_{12}\mathbf{b}_{11}, \mathbf{a}_{12}\mathbf{b}_{12}, \dots, \mathbf{a}_{13}\mathbf{b}_{11}, \dots$$

1. There exist $(b_2 - 1) + (r_1 - 1) \cdot b_2$ players that possess the element $\mathbf{a}_{11}\mathbf{b}_{ij}$ for some $i \in \{1, 2, \dots, b_2\}$ and $j \in \{1, 2, \dots, k_2\}$, since r_1 is the replication number of \mathcal{A} . Of these, $(r_1 - 1) \cdot 1$ players possess the first k_2 elements of the share, i.e., $\mathbf{a}_{11}\mathbf{b}_{11} \mathbf{a}_{11}\mathbf{b}_{12} \dots \mathbf{a}_{11}\mathbf{b}_{1k_2}$. If any of these players knows the ratios $\frac{\mathbf{a}_{12}}{\mathbf{a}_{11}}, \frac{\mathbf{a}_{13}}{\mathbf{a}_{11}}, \dots$, then they could construct the entire share of P_1 .
2. Note that for $j \neq 1$, any of the $b_2 - 1$ players with shares

$$\begin{array}{cccc} \mathbf{a}_{11}\mathcal{B}_2 & | & \mathbf{a}_{12}\mathcal{B}_2 & | & \dots & | & \mathbf{a}_{1k_1}\mathcal{B}_2, \\ \mathbf{a}_{11}\mathcal{B}_3 & | & \mathbf{a}_{12}\mathcal{B}_3 & | & \dots & | & \mathbf{a}_{1k_1}\mathcal{B}_3, \\ & & & & \vdots & & \\ \mathbf{a}_{11}\mathcal{B}_{b_2} & | & \mathbf{a}_{12}\mathcal{B}_{b_2} & | & \dots & | & \mathbf{a}_{1k_1}\mathcal{B}_{b_2} \end{array}$$

knows these ratios.

Therefore, only two players — one from the $r_1 - 1$ players possessing $\mathbf{a}_{11}\mathbf{b}_{11}$ and one from the $b_2 - 1$ players possessing $\frac{\mathbf{a}_{12}}{\mathbf{a}_{11}}, \frac{\mathbf{a}_{13}}{\mathbf{a}_{11}}, \dots$ — can reconstruct the entire share of player P_1 , and hence, frame this player.

We try to address this problem by reducing the repetitive nature of shares of the participants. We shall do this by decreasing the size of each share, while retaining all the information that a player had in the previous construction (i.e., Equation (3.4)), thus increasing the number of players required to frame another player.

3.7.1 A Modified Scheme

Given two matrices \mathcal{A} and \mathcal{B} of the same dimension $r \times c$, we define the operation $\mathcal{A} \odot \mathcal{B}$ as the $r \times c$ matrix generated by position-wise products of elements of \mathcal{A} and \mathcal{B} , i.e., if

$$\mathcal{A} = \begin{pmatrix} \mathbf{a}_{11} & \mathbf{a}_{12} & \cdots & \mathbf{a}_{1c} \\ \vdots & & & \\ \mathbf{a}_{r1} & \mathbf{a}_{r2} & \cdots & \mathbf{a}_{rc} \end{pmatrix} \text{ and } \mathcal{B} = \begin{pmatrix} \mathbf{b}_{11} & \mathbf{b}_{12} & \cdots & \mathbf{b}_{1c} \\ \vdots & & & \\ \mathbf{b}_{r1} & \mathbf{b}_{r2} & \cdots & \mathbf{b}_{rc} \end{pmatrix}, \text{ then}$$

$$\mathcal{A} \odot \mathcal{B} = \begin{pmatrix} \mathbf{a}_{11}\mathbf{b}_{11} & \mathbf{a}_{12}\mathbf{b}_{12} & \cdots & \mathbf{a}_{1c}\mathbf{b}_{1c} \\ \vdots & & & \\ \mathbf{a}_{r1}\mathbf{b}_{r1} & \mathbf{a}_{r2}\mathbf{b}_{r2} & \cdots & \mathbf{a}_{rc}\mathbf{b}_{rc} \end{pmatrix}.$$

The operator \odot is well-behaved in the sense that it is commutative and respects scalar multiplication on integer-valued matrices.

Let $\pi : \{1, 2, \dots, b\} \rightarrow \{1, 2, \dots, b\}$ be a permutation. Given $i \in \{1, 2, \dots, b\}$ and $\pi(i) = j$, we define $\tilde{\pi} : \{1, 2, \dots, b\} \rightarrow \{1, 2, \dots, k\}$ as $\tilde{\pi}(i) = j \pmod{k}$, for any integer $k \leq b$. Now given BIBDs $\mathcal{A}_{b_1 \times k_1}$ and $\mathcal{B}_{b_2 \times k_2}$, we modify their Krönecker product by first choosing a permutation π_1 randomly from the set of all permutations over $\{1, 2, \dots, b_2\}$ and producing $\tilde{\pi}_1$. Then we produce $\tilde{\pi}_2, \tilde{\pi}_3, \dots, \tilde{\pi}_{k_1}$ by simple translations.

Next, we represent application of the function $\tilde{\pi}_l$ to the m^{th} block matrix (of size $b_2 \times k_2$) of block-row t in $\mathcal{A} \otimes \mathcal{B}$ by $\theta_{mt} = l$, and define matrix $N_{b_1 b_2 \times k_1 k_2} = (n_{ij})$ divided into blocks of size $b_2 \times k_2$ similarly as $\mathcal{A} \otimes \mathcal{B}$ such that

$$\begin{cases} n_{ij} = 1 & \text{if } \tilde{\pi}_l(i) = j \\ n_{ij} = 0 & \text{if otherwise} \end{cases},$$

where n_{ij} is the element in the i^{th} row and j^{th} column of the $(m, t)^{\text{th}}$ block matrix of M . Finally, the i^{th} row of matrix $(\mathcal{A} \otimes \mathcal{B}) \odot N$ produces the share of player P_i ($i \in \{1, 2, \dots, b_1 b_2\}$) by omitting the zeroes.

3.7.2 Example

Consider another example, where a $(4, 3, 2)$ -BIBD and a $(5, 4, 3)$ -BIBD over the points $\{1, 2, 3, 4\}$ and $\{22, 23, 24, 25, 26\}$ are represented by matrices \mathcal{A} and \mathcal{B} , respectively (note that $r_1 = 3, r_2 = 4$):

$$\mathcal{A} = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 3 & 4 & 1 \\ 4 & 1 & 2 \end{pmatrix}, \text{ and } M_{\mathcal{B}} = \begin{pmatrix} 22 & 23 & 24 & 25 \\ 23 & 24 & 25 & 26 \\ 24 & 25 & 26 & 22 \\ 25 & 26 & 22 & 23 \\ 26 & 22 & 23 & 24 \end{pmatrix}. \quad (3.12)$$

Then $b_1 = 4, b_2 = 5, k_1 = 3$ and $k_2 = 4; \tau_1 = 2$ and $\tau_2 = 2$ are the reconstruction numbers of \mathcal{A} and \mathcal{B} , respectively.

Modifying the matrix in Figure 3.2, as shown in Figures 3.3 and 3.4, we obtain a scheme for which it is no longer possible to reconstruct the secret of the scheme in Figure 3.4 from just two players (as was possible in the example in Section 3.5). In fact, the subsequent section (Section 3.7.3) provides an algorithm for secret reconstruction from this scheme using $\tau_1 + \tau_2$ players.

22	23	24	25	44	46	48	50	66	69	72	75
23	24	25	26	46	48	50	52	69	72	75	78
24	25	26	22	48	50	52	44	72	75	78	66
25	26	22	23	50	52	44	46	75	78	66	69
26	22	23	24	52	44	46	48	78	66	69	72
44	46	48	50	66	69	72	75	88	92	96	100
46	48	50	52	69	72	75	78	92	96	100	104
48	50	52	44	72	75	78	66	96	100	104	88
50	52	44	46	75	78	66	69	100	104	88	92
52	44	46	48	78	66	69	72	104	88	92	96
66	69	72	75	88	92	96	100	22	23	24	25
69	72	75	78	92	96	100	104	23	24	25	26
72	75	78	66	96	100	104	88	24	25	26	22
75	78	66	69	100	104	88	92	25	26	22	23
78	66	69	72	104	88	92	96	26	22	23	24
88	92	96	100	22	23	24	25	44	46	48	50
92	96	100	104	23	24	25	26	46	48	50	52
96	100	104	88	25	26	22	23	48	50	52	44
100	104	88	92	25	26	22	23	50	52	44	46
104	88	92	96	26	22	23	24	52	44	46	48

Figure 3.2: The matrix $\mathcal{A} \otimes \mathcal{B}$ is the Kröner product of \mathcal{A} and \mathcal{B} as in Equation (3.12), and is a secret sharing scheme with reconstruction number 2.

1	0	0	0	0	0	0	1	0	0	1	0
1	0	0	0	1	0	0	0	0	0	0	1
0	1	0	0	1	0	0	0	1	0	0	0
0	0	1	0	0	1	0	0	1	0	0	0
0	0	0	1	0	0	1	0	0	1	0	0
0	0	0	1	0	0	1	0	1	0	0	0
1	0	0	0	0	0	0	1	1	0	0	0
1	0	0	0	1	0	0	0	0	1	0	0
0	1	0	0	1	0	0	0	0	0	1	0
0	0	1	0	0	1	0	0	0	0	0	1
0	0	1	0	1	0	0	0	0	0	0	1
0	0	0	1	1	0	0	0	1	0	0	0
1	0	0	0	0	1	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	1	0	0
0	1	0	0	0	0	0	1	0	0	1	0
1	0	0	0	0	0	0	1	0	0	1	0
1	0	0	0	1	0	0	0	0	0	0	1
0	1	0	0	1	0	0	0	1	0	0	0
0	0	1	0	0	1	0	0	1	0	0	0
0	0	0	1	0	0	1	0	0	1	0	0

Figure 3.3: The matrix N , right-operated as $\odot N$ on the tensor design $\mathcal{A} \otimes \mathcal{B}$ in Figure 3.2

22	0	0	0	0	0	0	50	0	0	72	0	22	50	72
23	0	0	0	46	0	0	0	0	0	0	78	23	46	78
0	25	0	0	48	0	0	0	72	0	0	0	25	48	72
0	0	22	0	0	52	0	0	75	0	0	0	22	52	75
0	0	0	24	0	0	46	0	0	66	0	0	24	46	66
0	0	0	50	0	0	72	0	88	0	0	0	50	72	88
46	0	0	0	0	0	0	78	92	0	0	0	46	78	92
48	0	0	0	72	0	0	0	0	100	0	0	48	72	100
0	52	0	0	75	0	0	0	0	0	88	0	52	75	88
0	0	46	0	0	66	0	0	0	0	0	96	46	66	96
0	0	72	0	88	0	0	0	0	0	0	25	72	88	25
0	0	0	78	92	0	0	0	23	0	0	0	78	92	23
72	0	0	0	0	100	0	0	24	0	0	0	72	100	24
75	0	0	0	0	104	0	0	0	26	0	0	75	104	26
0	66	0	0	0	0	92	0	0	0	23	0	66	92	23
88	0	0	0	0	0	0	25	0	0	48	0	88	25	48
92	0	0	0	23	0	0	0	0	0	0	52	92	23	52
0	100	0	0	25	0	0	0	48	0	0	0	100	25	48
0	0	88	0	0	26	0	0	50	0	0	0	88	26	50
0	0	0	96	0	0	23	0	0	44	0	0	96	23	44

Figure 3.4: The matrix on the left is $(\mathcal{A} \otimes \mathcal{B}) \odot N$, and the one on the right is the share distribution scheme obtained from this operation, as described in Section 3.7.1.

3.7.3 Secret Reconstruction for the Modified Scheme

1. Choose a player P_i^m (which is the i^{th} player in the m^{th} row-block of $\mathcal{A} \otimes \mathcal{B}$, or the $((m-1)b_2 + i)^{\text{th}}$ player from the top), for any $m \in \{1, 2, \dots, b_1\}$ and $i \in \{1, 2, \dots, b_2\}$.
2. Consider elements $a_{mt}b_{ij}$ in the share of player P_i^m , i.e., $\theta_{mt} = l$ and $\tilde{\pi}^l(i) = j$. For such an element $a_{mt}b_{ij}$, set $y = b_{ij}$ (note that the value $y \in \{y_1, y_2, \dots, y_{v_2}\}$ is not known, but the positions at which the matrix \mathcal{B} contains elements $b_{\hat{i}\hat{j}} = y$ is known).
3. Construct set $\mathcal{S}_y := \left\{ \hat{l} : \left(\tilde{\pi}^{\hat{l}}(\hat{i}) = \hat{j} \right) \wedge (b_{\hat{i}\hat{j}} = y) \right\}$. By Theorem 3.4, for a maximal set \mathcal{S}_y (if not, then another value y may be chosen by selecting a different element $a_{m't'}b_{i'j'}$) the set

$$\begin{aligned}
 \{a_{\hat{m}\hat{t}} & : a_{\hat{m}\hat{t}}b_{\hat{i}\hat{j}} \in \text{the share of player } P_{\hat{i}}^{\hat{m}} \text{ such that } b_{\hat{i}\hat{j}} = y\} \\
 & = \{x_1, x_2, \dots, x_{v_1}\}
 \end{aligned}$$

is the set of all values in \mathcal{A} . Observe that this requires τ_1 players.

4. Construct matrix \mathcal{A} , since the positions of all values x_1, x_2, \dots, x_{v_1} in this matrix are now known.
5. Compute $b_{i'j'}$ for $a_{m't'}b_{i'j'} \in$ share of player $P_{i'}^{m'}$ using the known values $a_{m't'}$ until all values y_1, y_2, \dots, y_{v_2} are known. Observe that this requires τ_2 more players.
6. Construct matrix \mathcal{B} , since the positions of all values y_1, y_2, \dots, y_{v_2} in this matrix are now known.
7. Compute $\mathcal{A} \otimes \mathcal{B}$ from the two known matrices.

Thus, framing any player is not possible for just two other participants, and requires a much larger coalition of $\tau_1 + \tau_2$ players.

3.8 Graphical Representation and Proof of Existence of Permutations

Matching in Bipartite Graphs

We shall now use the concepts of matchings and covers to analyze the frameproofness of the secret sharing scheme. We leverage bipartite graphs to model the relationships between players and the secrets they can access, ensuring that coalitions cannot frame other players. We use the graph theoretical concepts discussed in section 2.3 for this purpose.

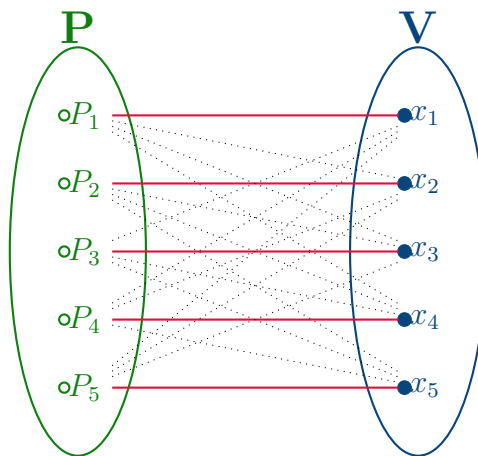


Figure 3.5: A bipartite graph for the tensor design \mathcal{B} defined in Section 3.7.2 with 5 players and 5 points. Each edge (P_i, x_j) denotes the inclusion of point x_j in the share of player P_i . The collection of red edges shows one possible maximal matching for the graph.

Definition 3.2 A bipartite graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is said to induce a tensor design \mathcal{B} if

- the vertex set $\mathcal{V} = \mathbf{P} \sqcup \mathbf{V}$ the disjoint union of the set of players $\mathbf{P} = \{P_1, \dots, P_b\}$ and the set of points $\mathbf{V} = \{x_1, \dots, x_v\}$ of \mathcal{B} ;
- the edge set is the collection $\bigcup_{\substack{i \in [b] \\ j \in [v]}} \{(P_i, x_j) : x_j \in \text{share of } P_i\}$.

Theorem 3.4 Given a bipartite graph \mathcal{G} inducing a tensor design \mathcal{B} , and given subsets $\delta(P_i) \subseteq N(P_i)$ of size s ,

- (i) If $\bigcup_{i \in [b]} \delta(P_i) = \mathbf{V}$, then reconstruction of the modified scheme $(\mathcal{A} \otimes \mathcal{B})_{\text{modified}}$ is possible.
- (ii) If $s \geq 1$, then (i) holds.

Proof: Assuming the usual notations for a tensor design, it is clear that in \mathcal{G} ,

$$\begin{aligned} |N(x_j)| &= r \quad \forall x_j \in \mathbf{V} \\ |N(P_{i_1} \cap P_{i_2})| &= \lambda \end{aligned} \tag{3.13}$$

From Equation (3.13) and the inclusion-exclusion principle,

$$|N(\{x_{i_1}, \dots, x_{i_m}\})| \geq m(r - \lambda)$$

Since $r \geq \lambda$, Hall's theorem (Theorem 2.11) implies \mathcal{G} has a matching of size v , i.e., $\bigcup_{i \in [b]} \delta(P_i) = \mathbf{V}$. Thus, (i) holds by the reconstruction algorithm in Section 3.7.3.

Now choose $\delta(P_i)$ such that each subset contains at least one point matched with P_i in this matching, so that (ii) holds. This proves the theorem. \square

3.9 Conclusions and Future Work

This chapter presents a significant advancement in the field of secret sharing by introducing a novel combinatorial design that surpasses existing methods in terms of efficiency, security, and flexibility. By generalizing the domain of distribution designs to integer rings, we achieve enhanced share repairability and simplified secret reconstruction. The integration of frameproofness significantly strengthens the scheme's resistance to malicious attacks, making it highly

suitable for sensitive applications. The proposed construction based on a simple Krönecker product offers a scalable and efficient approach to combining multiple secret sharing systems while preserving their individual properties.

We have thus, first generalized the concept of combinatorial RTS and then improved our secret sharing scheme by producing a frameproof one. Our findings underscore the potential of combinatorial designs for developing robust and secure secret sharing schemes tailored to diverse IoT applications. The proposed framework provides a solid foundation for future research in this area, with opportunities to explore further optimizations, new applications, and the integration of additional cryptographic primitives.

4 | Applications to IoT and Verifiability

4.1 Secret Sharing Schemes and the Internet of Things

The Internet of Things (IoT) refers to a network of interconnected devices, objects, and systems that are embedded with sensors, software, and other technologies to collect and exchange data over the internet. These devices can range from everyday objects such as smart home appliances and wearable devices to industrial machines and infrastructure components. IoT enables these devices to communicate with each other and with centralized systems, allowing for automation, data analysis, and improved efficiency in various domains such as healthcare, transportation, agriculture, and smart cities. Verifiable secret sharing schemes play a crucial role in ensuring the security, privacy, and integrity of sensitive data transmitted and stored by IoT devices. In collaborative IoT applications where multiple entities need to work together while preserving data privacy, verifiable secret sharing schemes can facilitate secure collaboration without compromising sensitive information. (Geng et al., 2022) proposes a privacy-preserving implementation of a VSS scheme, where information is split and encrypted to protect sensitive data during transmission. (Rehman, Saba, Haseeb, Larabi Marie-Sainte, & Lloret, 2021) also highlights the significance of verifiable secret sharing schemes in IoT, particularly in healthcare scenarios, where the protection of patient data is paramount. These schemes play a critical role in enhancing security, privacy, trust, and data integrity in IoT-based e-health systems, contributing to the overall reliability and effectiveness of healthcare applications. (Fu, Ren, Feng, Zhang, & Qin, 2021) proposes a non-interactive and secure data aggregation scheme that utilizes additive secret sharing to share data in two parts before masking these shared values, ensuring the data privacy of mobile users.

Secret sharing schemes can be used to distribute the security key amongst numerous devices in an IoT system, ensuring that no single device has access to the entire key. They are also lightweight and require less computational power compared to other cryptographic elements. Additionally, their ability to detect and prevent attacks that attempt to modify or delete parts of the secret is particularly important in IoT applications where security is critical.

For example, (Rehman et al., 2021) proposes an AI heuristic decision algorithm, utilizing a best-first search (BFS) approach. It effectively balances energy load and reduces communication

overhead in smart healthcare technologies. The utilization of homomorphic secret sharing in IoT-based e-health applications provides various advantages in terms of privacy and security. It securely distributes secret pairs among medical nodes, ensuring the confidentiality of sensitive health data during transmission and storage within the network. This is achieved by encrypting data through homomorphic secret sharing, thereby preventing unauthorized access to medical data. Access to medical records is limited to authorized entities possessing the necessary secret keys to decrypt and utilize the shared data. Thus, the incorporation of homomorphic secret sharing adds an extra layer of protection against unauthorized modifications or alterations to medical records. A generalization of this scheme to multiple levels—possibly to combine data between different hospitals or chains of healthcare providers, different states within a country, or even different countries—can be easily achieved through the Krönecker product of the individual schemes used by each hospital system. The fields on which these schemes are based provide a perfect foundation for the homomorphism, which can be easily maintained by the integer ring over which the Krönecker product is then defined.

A frameproof tensor product of multiple distribution designs can be distinctly useful for lightweight IoT applications, as it allows for a multi-level or multi-system secret sharing scheme IoT implementation in a secure and efficient manner, while detecting and preventing any attempt to modify or delete parts of the secret data. This approach ensures that even if some levels are compromised, the overall security of the system(s) remains intact.

The wide range of applicability of our generalizations can be further seen in, say, the management of massive data, such as (Fu et al., 2021), which proposes a non-interactive approach for IoT data aggregation that utilizes additive secret sharing, addressing numerous challenges including privacy concerns, security risks, high communication overhead, and user interaction. The additive secret sharing effectively masks the original data, preventing malicious analysis by the servers. The scheme also supports offline mobile users, maintains privacy, and provides efficient algorithms for result verification. However, (Fu et al., 2021) only splits the secret between two servers at a time. A frameproof tensor product can be smoothly applied in this context for connecting a large number of such systems, due to the underlying fields over which the secrets are split between servers in individual systems, as well as the generalized integer ring over which the tensor product is then defined.

Figure 4.1 shows an application of tensor design in multi-system IoT. We draw the reader's

attention to the applicability of our results from chapters 3 and 4 to secret sharing applications on the Internet of Things, especially in a secure, lightweight context.

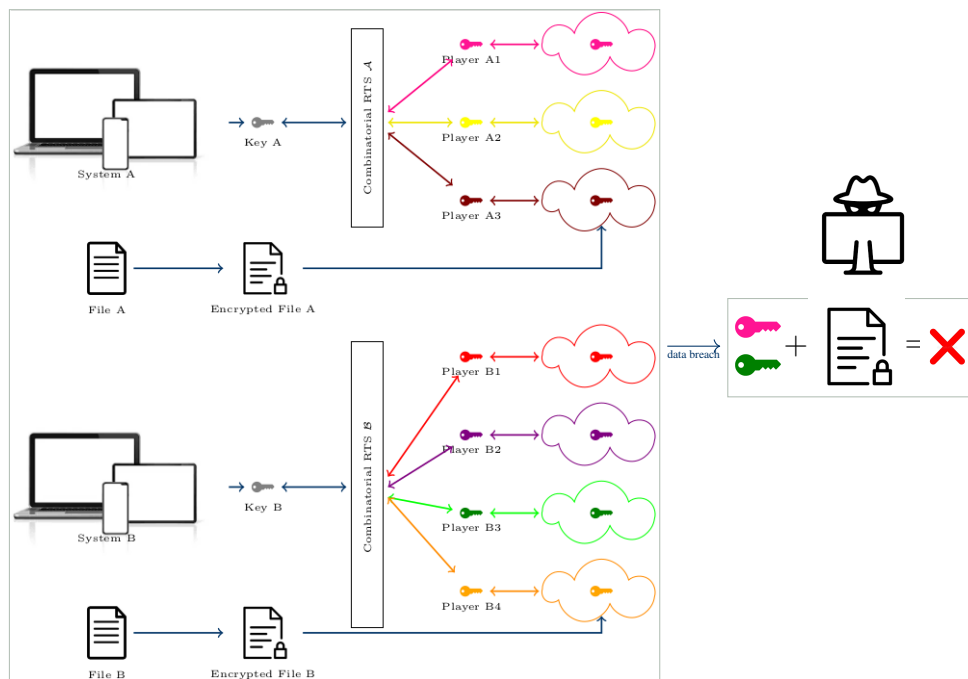


Figure 4.1: An application of the tensor product of repairable threshold schemes from Chapter 3 in multi-system IoT, where each system (say, a single hospital) may possess a separate RTS for sharing its own secret key, while multiple systems (say, a chain of hospitals) may share their individual secrets to non-colluding cloud storage providers through a tensor product of the individual schemes.

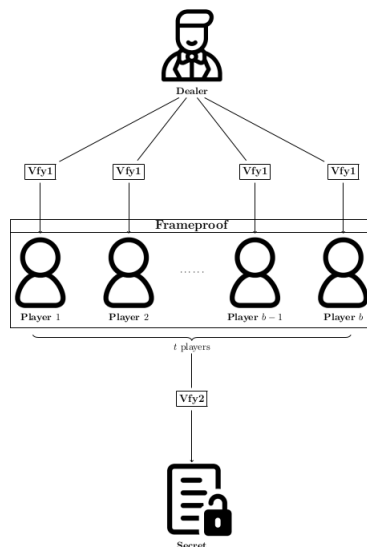
4.2 Vulnerabilities in Communication Networks

Some models of vulnerability and attacks/malicious behaviors are studied in detail in this domain. We briefly identify them here. A detailed discussion may be found in Chapters 6 and 7.

- **Share Distribution Stage:** Anomalies may be introduced during data transfer from the dealer to players.
- **Framing Dynamics:** There may be risks of players framing other players.
- **Malicious Share Insertion:** There may also be threats of false share contributions during the secret reconstruction phase.

Section 4.3 addresses one such vulnerability: anomaly due to erroneous share distribution. One can use error-correcting codes (Poli, 1985; Yao & Cheng, 1986; Dong, Mani, & Zhao, 2023) as

well as repair techniques described in (Kacsmar & Stinson, 2019) to reconstruct faulty shares. Further chapters (Chapters 5 and 6) explore these concepts in detail.



4.3 Verifiability in Secret Sharing

Verifiable secret sharing schemes play a crucial role in ensuring the security, privacy, and integrity of sensitive medical data transmitted and stored by IoT devices. They are essential for maintaining the security and privacy of medical data in IoT-based healthcare systems. These schemes enable the distribution of secret pairs among peer medical sensors in a secure manner, ensuring that sensitive information is protected from unauthorized access and malicious nodes. They also help IoT devices establish trust amongst each other, as it is ensured that only authorized devices have access to the shared secrets. Moreover, verifiable secret sharing schemes help maintain the integrity of medical data transmitted and stored by IoT devices and prevent data tampering and unauthorized modifications, ensuring the reliability of the information exchanged between devices.

Existing Verifiable Secret Sharing (VSS) schemes are based on several foundational concepts in cryptography and distributed computing. The security of VSS schemes often relies on one-way functions, which are easy to compute in one direction but hard to invert. This property is used to create verification data that participants can use to check the validity of their shares. Many VSS schemes utilize homomorphic functions, which allow certain operations to be performed on the shares without needing to reconstruct the secret. For instance, Feldman’s VSS scheme (Feldman,

1987) employs a homomorphic one-way function to verify the consistency of shares. This property is crucial for ensuring that participants can validate their shares without needing to communicate with others. Some other VSS schemes incorporate zero-knowledge proofs to allow participants to prove that they possess a valid share without revealing any information about the share itself. This is particularly useful in scenarios where privacy and confidentiality are paramount. Likewise, many VSS schemes leverage public key cryptography to facilitate secure communication and verification.

Given the resource constraints of many IoT devices, existing VSS schemes are increasingly focused on efficiency in terms of computation and communication. This includes minimizing the amount of verification data required and reducing the computational overhead associated with share generation and verification. Many modern VSS schemes utilize cryptographic hash functions to ensure the integrity and authenticity of shares. Hash functions can be used to create compact representations of shares that can be easily verified.

4.4 Lightweight Share Verification

Cheater detection in Verifiable Secret Sharing (VSS) ensures that the secret can only be reconstructed by authorized shareholders who possess valid shares. This integrity is crucial in scenarios where the secret is sensitive or valuable, as it prevents malicious players from manipulating the reconstruction process to gain unauthorized access to the secret, or to maliciously sabotage the reconstruction of secrets without making any other gain for themselves. Since shareholders must trust that the shares they receive and use for reconstruction are legitimate, cheater detection mechanisms provide a way to identify and exclude dishonest participants from the reconstruction process. This is particularly important in environments where shareholders may not have prior relationships or trust established. In addition, identifying coalitions of dishonest shareholders attempting to reconstruct the secret using fake or manipulated shares is facilitated, allowing the VSS scheme to maintain its security. There is no doubt that the ability to detect cheaters enhances the overall robustness of the secret sharing scheme against various attacks, including those from insiders who may attempt to compromise the system. This is especially relevant in cloud computing environments. In fact, simply knowing that there are mechanisms in place for cheater detection can deter potential dishonest behavior amongst various participants. For these reasons, cheater detection is a critical component of VSS schemes,

as it ensures the security, integrity, and trustworthiness of the secret sharing process, making it suitable for applications in sensitive areas such as finance, healthcare, and cloud computing.

4.5 Existing Verification Protocols

One approach to achieving verifiability is through homomorphic commitment schemes, such as Benaloh's scheme (Benaloh, 1986). These allow shareholders to verify that all shares are collectively consistent without revealing the secret. However, this method requires interactive proofs to ensure the dealer's integrity, which can complicate the process and make it less practical. Another method by (Harn & Lin, 2009) involves verifying the coherence of shares by comparing the secrets reconstructed from different subsets of players. If all subsets yield the same secret, it indicates no cheating has occurred. This method requires a coalition of players larger than the threshold to effectively detect cheaters, which can be a limitation in smaller groups.

The verification algorithms of (Cafaro & Pellè, 2018) are designed to be space-efficient, meaning they do not require the storage of public data for verification, which reduces the overhead typically associated with secret sharing schemes. The proposed schemes can be used in conjunction with arbitrary secret sharing schemes and provide mechanisms for detecting cheaters among shareholders. Consequently, the design emphasizes robustness against cheaters by implementing verification routines that ensure the legality of shares independently from the secret they are generated from, unlike traditional homomorphic commitment schemes.

The Delegated Proof of Secret Sharing (DPoSS) consensus protocol proposed by (Geng et al., 2022) introduces several mathematically grounded contributions that address the challenges inherent in IoT environments. It optimizes the consensus process by leveraging secure multiparty computation (MPC) techniques (Luo, Deng, Wu, & Wang, 2019; Zhong, Sang, Zhang, & Xi, 2019), specifically through the use of Shamir's Secret Sharing (SSS) (Shamir, 1979). The protocol employs a randomized selection algorithm to elect nodes for block packing, which can be mathematically represented as a uniform distribution over the set of eligible nodes. Let N be the total number of nodes, and let $S \subseteq N$ be the subset of nodes eligible for selection. The probability $P(i)$ of node i being selected is given by $P(i) = \frac{1}{|S|} \quad \forall i \in S$. This ensures that each node has an equal chance of being selected, thereby promoting fairness and reducing

the risk of centralization, as no single node or small group of nodes can dominate the selection process. Further clarification of this process may be found in (Geng et al., 2022).

DPoSS incorporates verifiable secret sharing (VSS) by splitting the secret s into shares s_1, s_2, \dots, s_n using a polynomial $f(x)$ of degree $k - 1$ such that $f(0) = s$. Each share s_i is computed as $s_i = f(x_i)$ for distinct x_i values. The reconstruction of the secret requires at least k shares, ensuring that any coalition of fewer than k nodes cannot derive any information about s . This framework provides a robust mechanism for protecting sensitive data in the IoT context. Furthermore, the authors propose a modular architecture that allows for the integration of various secret sharing schemes, which can be mathematically represented by defining a set of secret sharing functions $\mathcal{F} = \{f_1, f_2, \dots, f_m\}$, where each function f_j corresponds to a different secret sharing scheme. The protocol can dynamically select f_j based on the specific requirements of the application, thus enhancing its versatility and efficiency across diverse IoT environments.

In particular, the key sharing protocol proposed in (Geng et al., 2022) incorporates VSS. The verification process involves a commitment phase in which, the dealer commits to the polynomial $f(x)$ by sending a commitment $C = \text{Commit}(f(x)) = (f(0), f(1), \dots, f(k - 1))$ to the nodes. This commitment can be done using cryptographic techniques such as hash functions or homomorphic encryption. It also involves a share verification phase in which, each node that receives a share s_i can verify it by checking if $s_i = f(x_i)$. If the share does not match the polynomial evaluation, the node can reject it and request a new share. The protocol ensures that no information about the secret is revealed unless k shares are combined, and that nodes can verify the correctness of the shares they receive.

The consensus protocols of (Geng et al., 2022) based on Verifiable Random Functions (VRFs) (which are a cryptographic primitive that produce a pseudorandom output from a given input, along with a proof that the output was generated correctly) ensure that the output appears random to anyone who does not know the secret key, allows any entity to verify that the output was generated correctly from the input and the secret key, and guarantees that for each input, the output is unique. Let K_s be the secret key and K_v be the verification key. The VRF consists of a key generation algorithm that generates a key pair (K_s, K_v) using a secure key generation algorithm, an evaluation algorithm such that for an input X , the output is $(Y, \pi) = \text{VRF_Eval}(K_s, X)$ (Y is the pseudorandom output, and π is the proof of correctness), and a verification algorithm, which given (Y, π) , checks if $\text{VRF_Verify}(K_v, X, Y, \pi) \rightarrow \text{True/False}$.

While the DPoSS protocol presents several advantages, it also has some drawbacks that can impact its implementation and performance in practical scenarios. DPoSS relies on the distribution of secret shares among nodes, which can lead to increased communication overhead, especially in large networks. The effectiveness of the secret sharing scheme is contingent upon the threshold k . If the number of malicious nodes exceeds $n - k$, the protocol's security can be compromised. The process of collecting shares, performing polynomial interpolation, and reconstructing the secret can introduce latency in reaching consensus. The time taken for nodes to communicate and verify shares can delay the block packing process, which may not be suitable for applications requiring real-time or near-real-time processing. Moreover, DPoSS assumes that a certain proportion (majority) of players are honest to function correctly. Additionally, the protocol's fault tolerance is inherently linked to the robustness of the underlying secret sharing scheme, which may not be sufficient in all scenarios.

Evidently, many existing VSS schemes face challenges such as requiring multiple rounds of communication (which can be inefficient and impractical in real-world applications), needing large numbers of polynomials or additional verification data, inadequate robustness against collusion among dishonest participants, etc. It is clear that despite significant progress in the field of verifiable secret sharing, challenges remain in terms of efficiency, robustness, and practicality. There is hence, a need for continued research to develop more effective and secure VSS schemes that can be applied in real-world scenarios.

4.6 An Improved Cheater Detection Algorithm

Let us first begin by recalling some ideas described in the previous chapter. Using two Shamir schemes on points x_1, \dots, x_{v_1} of a BIBD \mathcal{A} , and y_1, \dots, y_{v_2} of a BIBD \mathcal{B} , we construct a tensor design $\mathcal{F}(\mathcal{A}, \mathcal{B})$, which is frameproof. In short, we shall describe a verification protocol not based on hash functions. This protocol has a better computation complexity than standard hash-based verifiers and is moreover based on simple algebraic functions.

- The dealer chooses a (not very large) prime p such that none of $x_1, \dots, x_{v_1}, y_1, \dots, y_{v_2}$ are divisible by p , and declares p beforehand.
- He also produces a chart of inverse pairs $(a, a^{-1}) \forall a \in \mathbb{Z}_p$.
- The dealer then runs the share generation algorithm as described in Section 3.7.1.

- He then computes $\sum_{i,j} x_i^{-1} y_j^{-1} \pmod{p}$ for each player P , where the share of P consists of elements x_i from \mathcal{A} and y_j from \mathcal{B} .
- Finally, the dealer attaches this value to each share and distributes the shares to the participants.
- The secret reconstruction is done by an authorised collection of participants as in Section 3.7.3 and the verification can commence in this phase.

Complexity: In short, our computation complexity comes out to be $\mathcal{O}(n)$ – which is better than the previous $\mathcal{O}(n \log n)$.

Residue computation: at most $\mathcal{O}(\log^2 n)$.

Summation: $\mathcal{O}(n)$.

The storage space required is at most $p - 1$. However, the communication size increases.

4.7 Conclusion

In this chapter, we have demonstrated the broad applicability of the proposed scheme from Chapter 3 and the novel verification protocol in a variety of IoT contexts. This work underscores the critical need for ongoing research to develop more robust and secure VSS schemes suitable for real-world deployment. A promising avenue for future exploration involves in-depth analysis of specific use cases.

Cheater detection is a cornerstone of VSS, ensuring that only authorized shareholders with valid shares can reconstruct the secret. This integrity is paramount in scenarios where the secret carries significant value or sensitivity, as it safeguards against malicious attempts to manipulate the reconstruction process. We have discussed several existing VSS verification protocols that have explored various approaches. Homomorphic commitment schemes, such as Benaloh’s, provide one method. Harn and Lin’s technique focuses on verifying share coherence. Cafaro and Pellè’s algorithms prioritize space efficiency by eliminating the need for public data storage during verification, thereby reducing overhead. The DPoSS consensus protocol offers mathematically sound solutions to the unique challenges posed by IoT environments.

Building upon these foundations, we have introduced an improved cheater detection algorithm

that departs from traditional hash-based verification methods. This algorithm exhibits superior computational efficiency while relying on simple algebraic operations. Although it reduces storage requirements, it comes at the cost of increased communication overhead. This chapter lays the groundwork for further advancements in VSS, with a particular focus on enhancing security, efficiency, and practical implementation for IoT applications.

5.1 Introduction

The field of verifiable secret sharing schemes was introduced by Verheul et al. (Verheul & van Tilborg, 1997) and has evolved over time, incorporating well-known earlier examples by Feldman (Feldman, 1987) and Pedersen (Pedersen, 1991) that necessitated verifiability. Stinson made advancements in combinatorial design-based secret sharing schemes in 2004 (Stinson, 2004). Desmedt et al. introduced the concept of frameproofness in 2021 (Desmedt et al., 2021), while recent research by Sehrawat et al. in 2021 (V. S. Sehrawat et al., 2021) focuses on LWE-based access structure hiding verifiable secret sharing with malicious-majority settings. Furthermore, Roy et al. (B. K. Roy & Roy, 2023) combined the concepts of repairable threshold schemes by Stinson et al. and frameproofness by Desmedt et al. in 2023, to develop extendable tensor designs built from balanced incomplete block designs, and also presented a frameproof version of their design. This chapter explores ramp-type verifiable secret sharing schemes, and the application of hidden access structures in such cryptographic protocols. Inspired by Sehrawat et al.'s access structure hiding scheme, we develop an ϵ -almost access structure hiding scheme, which is verifiable as well as frameproof. We detail how the concept of ϵ -almost hiding is important for incorporating ramp schemes, thus making a fundamental generalisation of this concept.

Beginning with the introduction of various important types of secret sharing schemes such as VSS schemes, RTSs, BIBDs and access structure hiding schemes in Section 5.1, we define various notations, definitions and other preliminaries in Section 5.2. We introduce our modified concept of ϵ -almost access structure hiding ramp-type tensor designs in section 5.3, where we provide a background of the existing theory of extending tensor designs by Roy et al. (B. K. Roy & Roy, 2023), as well as demonstrate various secret sharing properties (such as correctness, ϵ -correctness and computational secrecy for their tensor design schemes. We also recall the concept of frameproof tensor designs through an example and show that it is also applicable to our scheme, and detail an algorithm for access structure token generation according to our requirements. In Section 5.4, we state the main results of this chapter in the form of Theorems 5.3, 5.4, 5.5 and 5.6. Sections 5.5 and 5.6 present detailed proofs of these theorems. In Section 5.7, we enumerate a few applications of our results in the real world, and then conclude

in Section 5.8.

Concepts of extremal set theory are used to hide the access structure.

5.2 Preliminaries

Given a collection $\mathbf{P} = \{P_1, \dots, P_\ell\}$ of (say) players in a secret sharing scheme, we denote the power set of \mathbf{P} , i.e. the set of all subsets of \mathbf{P} , by $2^{\mathbf{P}}$. The closure of a subset $\mathbf{A} \in 2^{\mathbf{P}}$ is the set $cl(\mathbf{A}) := \{\mathbf{C} : \mathbf{C}^* \subseteq \mathbf{C} \subseteq \mathbf{P} \text{ for some } \mathbf{C}^* \in \mathbf{A}\}$. Given a security parameter ω , a function $\delta(\omega)$ is called *negligible* if for all $c > 0$, there exists an ω_0 such that $\delta(\omega) < 1/\omega^c$ for all $\omega > \omega_0$. Given a probability distribution X , the notation $\Pr[t \leftarrow X]$ denotes a sampling of t by the distribution X .

Definition 5.1 *Let $X = \{X_\lambda\}_{\lambda \in \mathbb{N}}$ and $Y = \{Y_\lambda\}_{\lambda \in \mathbb{N}}$ be collections of probability distributions (or ensembles) X_λ and Y_λ over $\{0, 1\}^{\kappa(\lambda)}$ for some polynomial $\kappa(\lambda)$. These two ensembles are polynomially or computationally indistinguishable if for every (probabilistic) polynomial-time algorithm \mathbf{D} , for all $\lambda \in \mathbb{N}$, and a negligible function δ ,*

$$|\Pr[t \leftarrow X_\lambda : \mathbf{D}(t) = 1] - \Pr[t \leftarrow Y_\lambda : \mathbf{D}(t) = 1]| \leq \delta(\lambda).$$

Assume that there exist positive integers θ , Θ and ℓ , where $\theta < \Theta \leq \ell$. A (θ, Θ, ℓ) -*ramp scheme* (Paterson & Stinson, 2013) involves a dealer selecting a secret and then distributing a share to each of ℓ players in a manner that fulfills the following criteria:

Reconstruction: Any subset of Θ players has the ability to collectively determine the secret using the shares they possess.

Secrecy: No subset of θ players is able to deduce any details regarding the secret.

The terms θ and Θ are referred to as the lower and upper thresholds of the scheme, respectively. For the sake of convenience, we shall refer to collections of players $\mathbf{C} \in 2^{\mathbf{P}}$ such that $\theta < |\mathbf{C}| < \Theta$ by the term *ramp collection*. In the event where $\Theta = \theta + 1$, the scheme is recognized as a (Θ, ℓ) -threshold scheme. In the context of such a Θ -threshold scheme, the problem of *share repairability* pertains to the identification of a secure protocol for restoring the lost share of a specific player ($P_i \in \mathbf{P}$). This process involves a certain subset of d players (excluding $P_i \in \mathbf{P}$)

engaging in message exchange amongst themselves and with $P_i \in \mathbf{P}$, with the objective of successfully repairing its share. The smallest integer d required to accomplish this task is known as the *repairing degree* of the scheme. If an honest-but-curious coalition of no more than $\Theta - 1$ players of a (Θ, ℓ) -threshold scheme combines all the information it holds (this includes their shares, as well as all messages that they send or receive during the protocol) and still obtains no information about the secret, then we say that it is a (Θ, ℓ, d) -repairable threshold scheme, or a (Θ, ℓ, d) -RTS.

Definition 5.2 Let $\mathbf{P} = \{P_1, \dots, P_\ell\}$ be a set of parties or players. A collection $\Gamma \subseteq 2^{\mathbf{P}}$ is monotone if $\mathbf{A} \in \Gamma$ and $\mathbf{A} \subseteq \mathbf{B}$ imply that $\mathbf{B} \in \Gamma$. An access structure $\Gamma \subseteq 2^{\mathbf{P}}$ is a monotone collection of non-empty subsets of \mathbf{P} . Sets in γ are called authorized, and sets not in Γ are called unauthorized.

Definition 5.3 For an access structure Γ , $\Gamma_0 = \{\mathbf{A} \in \Gamma : \mathbf{B} \not\subseteq \mathbf{A} \text{ for all } \mathbf{B} \in \Gamma \setminus \mathbf{A}\}$ is the family of minimal authorized subsets in Γ .

Definition 5.4 A computational secret sharing scheme with respect to an access structure Γ , security parameter ω , a set of ℓ polynomial-time parties or players $\mathbf{P} = \{P_1, \dots, P_\ell\}$, and a set of secrets \mathbf{K} , consists of a pair of polynomial-time algorithms (Share, Recon), where:

- Share is a randomized algorithm that gets a secret $k \in \mathbf{K}$ and access structure Γ as inputs, and outputs ℓ shares, $\{s_1^{(k)}, \dots, s_\ell^{(k)}\}$, of k , and
- Recon is a deterministic algorithm that gets as input the shares of a subset $\mathbf{A} \subseteq \mathbf{P}$, denoted by $\left\{s_i^{(k)}\right\}_{i \in \mathbf{A}}$, and outputs a string in \mathbf{K} ,

such that the following two requirements are satisfied:

1. (Perfect Correctness) for all secrets $k \in \mathbf{K}$ and every authorized collection $\mathbf{A} \in \Gamma$, it holds that: $\Pr \left[\text{Recon} \left(\left\{s_i^{(k)}\right\}_{i \in \mathbf{A}}, \mathbf{A} \right) = k \right] = 1$,
2. (Computational Secrecy) for every unauthorized collection $\mathbf{B} \notin \Gamma$ and all distinct secrets $k_1, k_2 \in \mathbf{K}$, it holds that the distributions $\left\{s_i^{(k_1)}\right\}_{i \in \mathbf{A}}$ and $\left\{s_i^{(k_2)}\right\}_{i \in \mathbf{A}} \in \mathbf{B}$ are computationally indistinguishable (with respect to ω).

Traditionally, secret sharing relies on honest participants. However, a *verifiable secret sharing* (VSS) scheme is also required to withstand active attacks, specifically:

- a dealer sending inconsistent or incorrect shares to some of the participants during the distribution protocol, and
- participants submitting incorrect shares during the reconstruction protocol.

The access structure hiding verifiable (computational) secret sharing scheme of (V. S. Seehrawat et al., 2021) defined below guarantees a relaxed definition of verifiability of shares of authorised collections of players even when a majority of the parties are malicious. Their scheme supports all monotone access structures, and its security — in particular, verifiability — relies on the hardness of the LWE problem.

Definition 5.5 *An access structure hiding verifiable (computational) secret sharing scheme with respect to an access structure Γ , security parameter ω , a set of ℓ polynomial-time parties or players $\mathbf{P} = \{P_1, \dots, P_\ell\}$, and a set of secrets \mathbf{K} , consists of two sets of polynomial-time algorithms, $(\text{HsGen}, \text{HsVer})$ and $(\text{VerShr}, \text{Recon}, \text{Ver})$, which are defined as follows:*

- *VerShr is a randomized algorithm that gets a secret $k \in \mathbf{K}$ and access structure Γ as inputs, and outputs ℓ shares, $\{s_1^{(k)}, \dots, s_\ell^{(k)}\}$, of k ,*
- *Recon is a deterministic algorithm that gets as input the shares of a subset $\mathbf{A} \subseteq \mathbf{P}$, denoted by $\left\{s_i^{(k)}\right\}_{i \in \mathbf{A}}$, and outputs a string in \mathbf{K} , and*
- *Ver is a deterministic Boolean algorithm that gets $\left\{s_i^{(k)}\right\}_{i \in \mathbf{A}}$ and a secret $k' \in \mathbf{K}$ as inputs, and outputs $b \in \{0, 1\}$,*

such that the following three requirements are satisfied:

1. (Perfect Correctness) *for all secrets $k \in \mathbf{K}$ and every authorized collection $\mathbf{A} \in \Gamma$, it holds that: $\Pr \left[\text{Recon} \left(\left\{s_i^{(k)}\right\}_{i \in \mathbf{A}}, \mathbf{A} \right) = k \right] = 1$.*
2. (Computational Secrecy) *for every unauthorized collection $\mathbf{B} \notin \Gamma$ and all distinct secrets $k_1, k_2 \in \mathbf{K}$, it holds that the distributions $\left\{s_i^{(k_1)}\right\}_{i \in \mathbf{A}}$ and $\left\{s_i^{(k_2)}\right\}_{i \in \mathbf{A}} \in \mathbf{B}$ are computationally indistinguishable (with respect to ω).*
3. (Computational Verifiability) *Every authorized collection $\mathbf{A} \in \Gamma$ can use Ver to verify whether its set of shares $\left\{s_i^{(k)}\right\}_{i \in \mathbf{A}}$ is consistent with a given secret $k \in \mathbf{K}$. Formally, for a negligible function δ , it holds that:*

- If all shares $s_i^{(k)} \in \{s_i^{(k)}\}_{i \in \mathbf{A}}$ are consistent with the secret k , then

$$\Pr \left[\mathbf{Ver} \left(k, \{s_i^{(k)}\}_{i \in \mathbf{A}} \right) = 1 \right] = 1 - \delta(\omega)$$

- If any share $s_i^{(k)} \in \{s_i^{(k)}\}_{i \in \mathbf{A}}$ is inconsistent with the secret k , then

$$\Pr \left[\mathbf{Ver} \left(k, \{s_i^{(k)}\}_{i \in \mathbf{A}} \right) = 0 \right] = 1 - \delta(\omega).$$

- HsGen is a randomized algorithm that gets \mathbf{P} and Γ as inputs, and outputs ℓ access structure tokens $\{\mathcal{U}_1^{(\Gamma)}, \dots, \mathcal{U}_\ell^{(\Gamma)}\}$, and
- HsVer is a deterministic algorithm that gets as input the access structure tokens of a subset $\mathbf{A} \subseteq \mathbf{P}$ (denoted $\{\mathcal{U}_i^{(\Gamma)}\}_{i \in \mathbf{A}}$), and outputs $b \in \{0, 1\}$,

such that the following three requirements are satisfied:

1. (Perfect completeness) Every authorized collection of parties $\mathbf{A} \in \Gamma$ can identify itself as a member of the access structure Γ , i.e. $\Pr \left[\mathbf{HsVer} \left(\{\mathcal{U}_i^{(\Gamma)}\}_{i \in \mathbf{A}} \right) = 1 \right] = 1$.
2. (Perfect soundness) Every unauthorized collection of parties $\mathbf{B} \notin \Gamma$ can identify itself to be outside of the access structure Γ , i.e. $\Pr \left[\mathbf{HsVer} \left(\{\mathcal{U}_i^{(\Gamma)}\}_{i \in \mathbf{B}} \right) = 0 \right] = 1$.
3. (Statistical hiding) For all access structures $\Gamma, \Gamma' \subseteq 2^{\mathbf{P}}$ where $\Gamma \neq \Gamma'$, and for all unauthorised collections $\mathbf{B} \notin \Gamma, \Gamma'$,

$$\left| \Pr \left[\Gamma \mid \{\mathcal{U}_i^{(\Gamma)}\}_{i \in \mathbf{B}}, \{s_i^{(k)}\}_{i \in \mathbf{B}} \right] - \Pr \left[\Gamma' \mid \{\mathcal{U}_i^{(\Gamma)}\}_{i \in \mathbf{B}}, \{s_i^{(k)}\}_{i \in \mathbf{B}} \right] \right| = 2^{-\omega}.$$

5.3 ϵ -Almost Access Structure Hiding Ramp-Type Tensor Designs

So far, access structure hiding and related concepts have been primarily discussed in the context of threshold schemes. In such schemes, there exists a deterministic algorithm to determine whether an authorized set of players can recover the secret and whether an unauthorized set gains no information about the secret.

We integrate the novel access structure hiding technique from (V. S. Sehwat et al., 2021) into the tensor design obtained by extending BIBDs, as introduced in Chapter 3.

Since the scheme in Chapter 3 is a ramp scheme for both the non-frameproof and frameproof variants (defined below) of the tensor design, we introduce the new concept of an ϵ -almost access structure hiding ramp scheme in order to also tackle the intermediate case(s) generated by ramp bounds.

Definition 5.6 Consider a (θ, Θ, ℓ) -ramp scheme, so that its access structure Γ is characterised by the ramp bounds (θ, Θ) . For $\epsilon = (\epsilon_{\text{corr}}, \epsilon_1, \epsilon_2, \epsilon_3)$, an ϵ -almost access structure hiding (θ, Θ, ℓ) -ramp scheme with respect to a security parameter ω , a set of ℓ polynomial-time parties or players $\mathbf{P} = \{P_1, \dots, P_\ell\}$, and a set of secrets \mathbf{K} , consists of two sets of polynomial-time algorithms, $(\text{HsGen}, \text{HsVer})$ and $(\text{VerShr}, \text{Recon}, \text{Ver})$, which are defined as follows:

- *VerShr* is a randomized algorithm that gets a secret $k \in \mathbf{K}$ and the bounds θ, Θ as inputs, and outputs ℓ shares, $\{s_1^{(k)}, \dots, s_\ell^{(k)}\}$, of k ,
- *Recon* is a deterministic algorithm that gets as input the shares of a subset $\mathbf{A} \subseteq \mathbf{P}$, denoted by $\left\{s_i^{(k)}\right\}_{i \in \mathbf{A}}$, and outputs a string in \mathbf{K} , and
- *Ver* is a deterministic Boolean algorithm that gets $\left\{s_i^{(k)}\right\}_{i \in \mathbf{A}}$ and a secret $k' \in \mathbf{K}$ as inputs, and outputs $b \in \{0, 1\}$,

such that the following four requirements are satisfied:

1. (Perfect Correctness) for all secrets $k \in \mathbf{K}$ and every authorized collection \mathbf{A} such that $|\mathbf{A}| \geq \Theta$, it holds that: $\Pr \left[\text{Recon} \left(\left\{s_i^{(k)}\right\}_{i \in \mathbf{A}}, \mathbf{A} \right) = k \right] = 1$.
2. (ϵ_{corr} -Correctness) for all secrets $k \in \mathbf{K}$ and every ramp collection \mathbf{C} such that $\theta < |\mathbf{C}| < \Theta$, there exists $\epsilon_{\text{corr}} > 0$ such that: $\Pr \left[\text{Recon} \left(\left\{s_i^{(k)}\right\}_{i \in \mathbf{A}}, \mathbf{A} \right) = k \right] = \epsilon_{\text{corr}}$.
3. (Computational Secrecy) for every unauthorized collection \mathbf{B} with $|\mathbf{B}| \leq \theta$ and all distinct secrets $k_1, k_2 \in \mathbf{K}$, it holds that the distributions $\left\{s_i^{(k_1)}\right\}_{i \in \mathbf{A}}$ and $\left\{s_i^{(k_2)}\right\}_{i \in \mathbf{A}} \in \mathbf{B}$ are computationally indistinguishable (with respect to ω).
4. (Computational Verifiability) Every authorized collection \mathbf{A} such that $|\mathbf{A}| \geq \Theta$ can use *Ver* to verify whether its set of shares $\left\{s_i^{(k)}\right\}_{i \in \mathbf{A}}$ is consistent with a given secret $k \in \mathbf{K}$. Formally, for a negligible function δ , it holds that:

- If all shares $s_i^{(k)} \in \{s_i^{(k)}\}_{i \in \mathbf{A}}$ are consistent with the secret k , then

$$\Pr \left[\mathbf{Ver} \left(k, \{s_i^{(k)}\}_{i \in \mathbf{A}} \right) = 1 \right] = 1 - \delta(\omega)$$

- If any share $s_i^{(k)} \in \{s_i^{(k)}\}_{i \in \mathbf{A}}$ is inconsistent with the secret k , then

$$\Pr \left[\mathbf{Ver} \left(k, \{s_i^{(k)}\}_{i \in \mathbf{A}} \right) = 0 \right] = 1 - \delta(\omega).$$

- HsGen is a randomized algorithm that gets \mathbf{P} , θ and Θ as inputs, and outputs ℓ access structure tokens $\{\mathcal{U}_1^{(\Gamma)}, \dots, \mathcal{U}_\ell^{(\Gamma)}\}$, and
- HsVer is a deterministic algorithm that gets as input the access structure tokens of a subset $\mathbf{A} \subseteq \mathbf{P}$ (denoted $\{\mathcal{U}_i^{(\Gamma)}\}_{i \in \mathbf{A}}$), and outputs $b \in \{0, 1\}$,

such that the following six requirements are satisfied:

1. (Perfect completeness) Every authorized collection of parties \mathbf{A} such that $|\mathbf{A}| \geq \Theta$ can identify itself as a member of the access structure Γ , i.e. $\Pr \left[\mathbf{HsVer} \left(\{\mathcal{U}_i^{(\Gamma)}\}_{i \in \mathbf{A}} \right) = 1 \right] = 1$.
2. (ϵ_1 -Completeness) Every ramp collection of parties \mathbf{C} (where $\theta < |\mathbf{C}| < \Theta$) can almost always identify itself as a member of the access structure Γ , i.e. $\Pr \left[\mathbf{HsVer} \left(\{\mathcal{U}_i^{(\Gamma)}\}_{i \in \mathbf{A}} \right) = 1 \right] = 1 - \epsilon_1$.
3. (Perfect soundness) Every unauthorized collection of parties \mathbf{B} with $|\mathbf{B}| \leq \theta$ can identify itself to be outside of the access structure Γ , i.e. $\Pr \left[\mathbf{HsVer} \left(\{\mathcal{U}_i^{(\Gamma)}\}_{i \in \mathbf{B}} \right) = 0 \right] = 1$.
4. (ϵ_2 -Soundness) Every ramp collection of parties \mathbf{C} (where $\theta < |\mathbf{C}| < \Theta$) can almost always identify itself to be outside of the access structure Γ , i.e. $\Pr \left[\mathbf{HsVer} \left(\{\mathcal{U}_i^{(\Gamma)}\}_{i \in \mathbf{B}} \right) = 0 \right] = 1 - \epsilon_2$.
5. (Statistical hiding) For all ramp access structures $\Gamma \neq \Gamma'$ and for all unauthorised collections \mathbf{B} with $|\mathbf{B}| \leq \theta, \theta'$,

$$\left| \Pr \left[\Gamma \mid \{\mathcal{U}_i^{(\Gamma)}\}_{i \in \mathbf{B}}, \{s_i^{(k)}\}_{i \in \mathbf{B}} \right] - \Pr \left[\Gamma' \mid \{\mathcal{U}_i^{(\Gamma)}\}_{i \in \mathbf{B}}, \{s_i^{(k)}\}_{i \in \mathbf{B}} \right] \right| = 2^{-\omega}.$$

6. (ϵ_3 -Statistical Hiding) For all ramp access structures $\Gamma, \Gamma' \subseteq 2^{\mathbf{P}}$ where $\Gamma \neq \Gamma'$, and for all ramp collections \mathbf{C} such that $\theta < |\mathbf{C}| < \Theta$,

$$\left| \Pr \left[\Gamma \mid \left\{ \mathcal{U}_i^{(\Gamma)} \right\}_{i \in \mathbf{C}}, \left\{ s_i^{(k)} \right\}_{i \in \mathbf{C}} \right] - \Pr \left[\Gamma' \mid \left\{ \mathcal{U}_i^{(\Gamma)} \right\}_{i \in \mathbf{C}}, \left\{ s_i^{(k)} \right\}_{i \in \mathbf{C}} \right] \right| \leq \epsilon_3(\omega).$$

5.3.1 Tensor Design

Recall from Chapter 3 that if \mathcal{A} and \mathcal{B} are the share matrices generated by ramp schemes with respectively b_1 and b_2 blocks having shares of sizes k_1 and k_2 , and if \mathcal{A} and \mathcal{B} also denote the $b_1 \times k_1$ and $b_2 \times k_2$ matrices corresponding to the two schemes, then Krönecker product of $\mathcal{A} \otimes \mathcal{B}$ is

$$M = \begin{pmatrix} \mathbf{a}_{11}\mathcal{B} & \mathbf{a}_{12}\mathcal{B} & \dots & \mathbf{a}_{1k_1}\mathcal{B} \\ \mathbf{a}_{21}\mathcal{B} & \mathbf{a}_{22}\mathcal{B} & \dots & \mathbf{a}_{2k_1}\mathcal{B} \\ \vdots & & & \\ \mathbf{a}_{b_1 1}\mathcal{B} & \mathbf{a}_{b_1 2}\mathcal{B} & \dots & \mathbf{a}_{b_1 k_1}\mathcal{B} \end{pmatrix}. \quad (5.1)$$

If the share matrix \mathcal{A} is defined over the field \mathbb{F}_{p_1} and \mathcal{B} over the field \mathbb{F}_{p_2} for some primes p_1 and p_2 , then we define the scalar multiplication as the simple integer multiplication:

$$\begin{aligned} \mathbb{F}_{p_1} \times \mathbb{F}_{p_2} &\rightarrow \mathbb{Z} \\ \text{such that } (x_1, x_2) &\mapsto x_1 \cdot x_2. \end{aligned}$$

The reason behind taking such a multiplication is that the product elements are not distinguishable from integers. Therefore, M is a matrix over the integer ring \mathbb{Z} .

Theorem 5.1 (Reconstruction from Tensor Designs, Chapter 3) Consider a $(v_1, k_1, \lambda_1, b_1, r_1)$ -BIBD \mathcal{A} and a $(v_2, k_2, \lambda_2, b_2, r_2)$ -BIBD \mathcal{B} .

1. The matrix $\mathcal{A} \otimes \mathcal{B}$ produces a tensor design (over the integer ring \mathbb{Z}) for a (public) integer d such that there are no multiplicative collisions of the type $x_i(y_j + d) = x_k(y_l + d)$ for $(i, j) \neq (k, l)$.
2.
 - If $\gcd(x_1, x_2, \dots, x_{v_1}) = 1$;
 - if $\gcd(y_1, y_2, \dots, y_{v_2}) = 1$;

then \mathcal{A} and \mathcal{B} can be reproduced from a collection of players in the new scheme $\mathcal{A} \otimes \mathcal{B}$,

hence enabling share repair and secret reconstruction.

For the purpose of real-world implementation, we consider a prime power q , which is computed from p_1, p_2 and d such that it is sufficiently greater than all the elements in $\mathcal{A} \otimes \mathcal{B}$.

5.3.2 Secret Sharing Properties of $\mathcal{A} \otimes \mathcal{B}$

Since $\mathcal{A} \otimes \mathcal{B}$ is a (θ, Θ, ℓ) -ramp scheme, it clearly satisfies the following properties of Definition 5.6:

Perfect Correctness: From Lemmas 3.4–3.9 of Chapter 3, it is clear that $\mathcal{A} \otimes \mathcal{B}$ is a (θ, Θ, ℓ) -ramp scheme, for $\theta = (\tau_1 - 1)(\tau_2 - 1) + 1$ and $\Theta = \min \{(\tau_1 - 1)b_2 + 1, (\tau_2 - 1)b_1 + 1\}$.

Hence, any \mathbf{A} with $|\mathbf{A}| \geq \Theta$ can reconstruct the secret with probability 1,

$$\text{i.e. } \Pr \left[\text{Recon} \left(\left\{ s_i^{(k)} \right\}_{i \in \mathbf{A}}, \mathbf{A} \right) = k \right] = 1.$$

ϵ_{corr} -Correctness: Suppose $\theta < |\mathbf{C}| < \Theta$ and \mathbf{C} gets partial information about $\mathcal{A} \otimes \mathcal{B}$, i.e. it can reconstruct exactly one of \mathcal{A} and \mathcal{B}_d , say \mathcal{A} (respectively \mathcal{B}_d). Then it must guess the secret of the other factor, i.e. \mathcal{B}_d (respectively \mathcal{A}) uniformly at random at best, i.e. with probability $\frac{1}{p_2}$ (respectively $\frac{1}{p_1}$). Therefore, for all secrets $k \in \mathbf{K}$ and such a ramp collection \mathbf{C} , we denote $\epsilon_{\text{corr}} := \max \left\{ \frac{1}{p_1}, \frac{1}{p_2} \right\}$. Therefore, $\Pr \left[\text{Recon} \left(\left\{ s_i^{(k)} \right\}_{i \in \mathbf{A}}, \mathbf{A} \right) = k \right] \leq \epsilon_{\text{corr}}$.

Computational Secrecy: Consider an unauthorised collection \mathbf{B} , with $|\mathbf{B}| \leq \theta$ or $\theta < |\mathbf{B}| < \Theta$. Thus, \mathbf{B} gets no information about the secret, which means it must guess (at best) uniformly at random, the secrets of both the factors \mathcal{A} and \mathcal{B}_d of $\mathcal{A} \otimes \mathcal{B}$. Hence, given the access structure Γ , it holds for every unauthorised collection $\mathbf{B} \notin \Gamma$ and every pair of different secrets $k_1 \neq k_2$ in \mathbf{K} that the distributions $\left\{ s_i^{(k_1)} \right\}_{i \in \mathbf{B}}$ and $\left\{ s_i^{(k_2)} \right\}_{i \in \mathbf{B}}$ are computationally indistinguishable w.r.t. the parameter $\delta := \frac{1}{p_1 p_2}$, according to Definition 5.1.

5.3.3 Frameproofness

For the collection \mathbf{P} of all players in the scheme, (V. S. Sehwat et al., 2021) make the following claim regarding its frameproofness:

“...the share of each party P_i is sealed as a PRIM-LWE instance such that the lattice basis, A_i , used to generate it is known only to P_i . Since A_i is required to generate P_i 's share, it is infeasible for any coalition of polynomial-time parties $\mathbf{A} \subset \mathbf{P}$ to

compute the share of $P_i \in \mathbf{P} \setminus \mathbf{A}$ without solving the LWE problem.”

Furthermore, Chapter 3 shows that for the tensor design in Equation (5.1), only two players — one from the $r_1 - 1$ players possessing $\mathbf{a}_{11} \mathbf{b}_{11}$ and one from the $b_2 - 1$ players possessing $\frac{\mathbf{a}_{12}}{\mathbf{a}_{11}}, \frac{\mathbf{a}_{13}}{\mathbf{a}_{11}}, \dots$ — can reconstruct the entire share of player P_1 , and hence, frame this player. They address this problem by reducing the repetitive nature of shares of the participants — by decreasing the size of each share, while retaining all the information that a player had in the previous construction. In fact, the secret reconstruction for the modified scheme is then shown to require at $\tau_1 + \tau_2$ players. Additionally, Theorem 5.2 below ensures that $\mathcal{F}(\mathcal{A}, \mathcal{B})$ is simply a Θ -threshold scheme for $\Theta = \tau_1 + \tau_2$ (and not a ramp scheme like (AoB)).

Example

Consider an example, where matrix \mathcal{A} represents a $2 - (4, 3, 2)$ -BIBD and \mathcal{B} a $2 - (5, 4, 3)$ -BIBD over the points $\{1, 2, 3, 4\}$ and $\{1, 2, 3, 4, 5\}$, respectively (note that $r_1 = 3, r_2 = 4$), and $d = 21$. The Krönecker product tensor design obtained from these two matrices is represented by the matrix $\mathcal{A} \otimes \mathcal{B}$ as defined in Chapter 3:

22	23	24	25	44	46	48	50	66	69	72	75
23	24	25	26	46	48	50	52	69	72	75	78
24	25	26	22	48	50	52	44	72	75	78	66
25	26	22	23	50	52	44	46	75	78	66	69
26	22	23	24	52	44	46	48	78	66	69	72
44	46	48	50	66	69	72	75	88	92	96	100
46	48	50	52	69	72	75	78	92	96	100	104
48	50	52	44	72	75	78	66	96	100	104	88
50	52	44	46	75	78	66	69	100	104	88	92
52	44	46	48	78	66	69	72	104	88	92	96
66	69	72	75	88	92	96	100	22	23	24	25
69	72	75	78	92	96	100	104	23	24	25	26
72	75	78	66	96	100	104	88	24	25	26	22
75	78	66	69	100	104	88	92	25	26	22	23
78	66	69	72	104	88	92	96	26	22	23	24
88	92	96	100	22	23	24	25	44	46	48	50
92	96	100	104	23	24	25	26	46	48	50	52
96	100	104	88	25	26	22	23	48	50	52	44
100	104	88	92	25	26	22	23	50	52	44	46
104	88	92	96	26	22	23	24	52	44	46	48

On applying certain permutations on each block of $\mathcal{A} \otimes \mathcal{B}$ (and removing zeroes), we obtain a scheme that extends the BIBDs \mathcal{A} and \mathcal{B} , where it is no longer possible to reconstruct the secret from just two players. The full algorithm may be found in Chapter 3. The shares of players in

this version, which we shall denote here by $\mathcal{F}(\mathcal{A}, \mathcal{B})$, are:

$$\begin{pmatrix} 22 & 50 & 72 \\ 23 & 46 & 78 \\ 25 & 48 & 72 \\ 22 & 52 & 75 \\ 24 & 46 & 66 \\ 50 & 72 & 88 \\ 46 & 78 & 92 \\ 48 & 72 & 100 \\ 52 & 75 & 88 \\ 46 & 66 & 96 \\ 72 & 88 & 25 \\ 78 & 92 & 23 \\ 72 & 100 & 24 \\ 75 & 104 & 26 \\ 66 & 92 & 23 \\ 88 & 25 & 48 \\ 92 & 23 & 52 \\ 100 & 25 & 48 \\ 88 & 26 & 50 \\ 96 & 23 & 44 \end{pmatrix}$$

5.3.4 Secret Sharing Properties of $\mathcal{F}(\mathcal{A}, \mathcal{B})$

From Theorem 5.2 stated below, it is clear that $\mathcal{F}(\mathcal{A}, \mathcal{B})$ is a (θ, Θ, ℓ) -ramp scheme, for $\theta = \tau_1 + \tau_2$ and $\Theta = \min\{(\tau_1 - 1)b_2 + 1, (\tau_2 - 1)b_1 + 1\}$. Therefore, it clearly satisfies the following properties of perfect correctness for all authorised collections of players of size greater than Θ , ϵ_{corr} -correctness for ramp collections of players that are authorised, and computational secrecy for all unauthorised collections of players (irrespective of size), from Definition 5.6.

A complete explanation is very similar to that for $\mathcal{A} \otimes \mathcal{B}$ given in Section 5.3.2.

5.3.5 Graphical Representation

Definition 5.7 A bipartite graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is said to induce a tensor design \mathcal{B} if

- the vertex set $\mathcal{V} = \mathbf{P} \sqcup \mathbf{V}$ the disjoint union of the set of players $\mathbf{P} = \{P_1, \dots, P_b\}$ and the set of points $\mathbf{V} = \{x_1, \dots, x_v\}$ of \mathcal{B} , and
- the edge set is the collection $\bigcup_{\substack{i \in [b] \\ j \in [v]}} \{(P_i, x_j) : x_j \in \text{share of } P_i\}$.

Theorem 5.2 Given a bipartite graph \mathcal{G} inducing a tensor design \mathcal{B} , and given subsets $\delta(P_i) \subseteq N(P_i)$ of size s ,

- If $\bigcup_{i \in [b]} \delta(P_i) = \mathbf{V}$, then reconstruction of the modified scheme $\mathcal{F}(\mathcal{A}, \mathcal{B})$ is possible.
- If $s \geq 1$, then (i) holds.

5.3.6 Defining Access Structure Tokens

Consider first, the Krönecker product tensor design $\mathcal{A} \otimes \mathcal{B}$ as defined in Equation (5.1).

Let $\mathbf{a}_1, \dots, \mathbf{a}_{v_1} \in \mathbb{F}_{p_1}$ be the elements in \mathcal{A} and $\mathbf{b}_1, \dots, \mathbf{b}_{v_2} \in \mathbb{F}_{p_2}$ be the elements in \mathcal{B} . The access structure tokens for the share of each player are elements of $\mathbb{Z}_2^{v_1} \times \mathbb{Z}_2^{v_2}$, computed according to Algorithm 1.

Algorithm 1 HsGen: Access structure tokens for the tensor designs $\mathcal{A} \otimes \mathcal{B}$ and $\mathcal{F}(\mathcal{A}, \mathcal{B})$

```

 $\gamma \xleftarrow{\$} \text{Perm}(\{0, 1\}^{v_1} \times \{0, 1\}^{v_2}).$ 
for  $1 \leq i \leq b_1 b_2$  do:                                     //player  $P_i$ 
    for  $1 \leq j \leq v_1$  do:                                     //element  $\mathbf{a}_j$ 
         $\hat{U}_i^{(1, \Gamma)} \leftarrow (\omega_1, \dots, \omega_{v_1})$  such that  $\omega_j = 1$  if and only if element  $\mathbf{a}_j$  of  $\mathcal{A}$  occurs as
        a product  $\mathbf{a}_j \mathbf{b}_l$  in the share of  $P_i$ .
    end for
    for  $1 \leq l \leq v_2$  do:                                     //element  $\mathbf{b}_l$ 
         $\hat{U}_i^{(2, \Gamma)} \leftarrow (\omega_1, \dots, \omega_{v_2})$  such that  $\omega_l = 1$  if and only if element  $\mathbf{b}_l$  of  $\mathcal{B}$  occurs as
        a product  $\mathbf{a}_j \mathbf{b}_l$  in the share of  $P_i$ .
    end for
     $(\hat{U}_1^{(\Gamma)}, \dots, \hat{U}_{b_1 b_2}^{(\Gamma)}) \leftarrow \gamma(\hat{U}_1^{(1, \Gamma)} || \hat{U}_1^{(2, \Gamma)}, \dots, \hat{U}_{b_1}^{(1, \Gamma)} || \hat{U}_{b_2}^{(2, \Gamma)})$ . //permutation
end for

```

Logical Condition

From Algorithm 1, it is clear that the authorisation of a collection of players \mathbf{B} can be determined directly from the intermediate vectors $\hat{U}_i^{(1,\Gamma)}$ and $\hat{U}_i^{(2,\Gamma)}$ used to compute their access structure tokens. Consider the two logical statements P and Q :

$$\begin{aligned} P & : \mathbf{B} \in \Gamma \\ Q & : \left(\bigvee_{i \in \mathbf{B}} \hat{U}_i^{(1,\Gamma)} \text{ has Hamming weight} \geq \tau_1 \right) \wedge \left(\bigvee_{i \in \mathbf{B}} \hat{U}_i^{(2,\Gamma)} \text{ has Hamming weight} \geq \tau_2 \right). \end{aligned} \quad (5.2)$$

Then from the definition of $\hat{U}_i^{(1,\Gamma)}$ and $\hat{U}_i^{(2,\Gamma)}$, it is clear that $P \leftrightarrow Q$. The proceeding lemma easily follows from this observation:

Lemma 5.1 *Let Γ denote the access structure for the tensor design $\mathcal{A} \otimes \mathcal{B}$. Then there exist parameters θ and Θ such that Γ is fully characterised by the following three conditions on any collection of players $\mathbf{B} \in 2^{\mathcal{P}}$:*

1. *If $|\mathbf{B}| < \theta$, then $\mathbf{B} \notin \Gamma$.*
2. *If $\theta \leq |\mathbf{B}| < \Theta$, then \mathbf{B} may or may not belong to Γ , i.e. it may or may not be authorised.*
3. *If $|\mathbf{B}| \geq \Theta$, then $\mathbf{B} \in \Gamma$.*

Proof: The proof follows by checking which collections of players satisfy the condition Q . If τ_1 and τ_2 are the reconstruction numbers of \mathcal{A} and \mathcal{B} , respectively. Then from Lemmas 3.4 and 3.7 of Chapter 3, $\theta = (\tau_1 - 1)(\tau_2 - 1) + 1$. Also, from Lemmas 3.5, 3.6, 3.8 and 3.9 of Chapter 3, $\Theta = \min \{(\tau_1 - 1)b_2 + 1, (\tau_2 - 1)b_1 + 1\}$. \square

Further observe that the permutation γ in Algorithm 1 ensures that a collection of players \mathbf{B} of size $t < \Theta$ cannot simply examine their tokens and conclude (with probability 1) whether or not it is authorised.

5.4 Main results

Theorem 5.3 *Given a positive integer d that satisfies Theorem 5.1, consider the tensor designs $\mathcal{A} \otimes \mathcal{B}$ with ramp structure (θ, Θ, ℓ) , for a secret k , and shares $s_i^{(k)}$ for each player $P_i \in \mathcal{P}$.*

Then there exists an access structure token generation algorithm that makes $\mathcal{A} \otimes \mathcal{B}$ an ϵ -almost access structure hiding (θ, Θ, ℓ) -ramp tensor design.

Theorem 5.4 Given a positive integer d that satisfies Theorem 5.1, consider the tensor designs $\mathcal{F}(\mathcal{A}, \mathcal{B})$ with ramp structure (θ, Θ, ℓ) , for a secret k , and shares $s_i^{(k)}$ for each player $P_i \in \mathbf{P}$. Then there exists an access structure token generation algorithm that makes $\mathcal{F}(\mathcal{A}, \mathcal{B})$ an ϵ -almost access structure hiding (θ, Θ, ℓ) -ramp tensor design.

Theorem 5.5 The access structure hiding tensor design $\mathcal{A} \otimes \mathcal{B}$ is verifiable.

Theorem 5.6 The access structure hiding tensor design $\mathcal{F}(\mathcal{A}, \mathcal{B})$ is verifiable.

5.5 Proof of Theorems 5.3 and 5.4

Proof: [Proof of Theorem 5.3.] This is easily seen as the scheme $\mathcal{A} \otimes \mathcal{B}$ satisfies the six properties enumerated in Definition 5.6.

Completeness and ϵ_1 -completeness:

Case 1: $|\mathbf{A}| \geq \Theta$. Since the access structure tokens of any collection of size at least Θ always satisfy the logical condition (5.2), \mathbf{A} can simply check this condition and output 1. Therefore,

$$\Pr \left[\text{HsVer} \left(\left\{ \mathcal{U}_i^{(\Gamma)} \right\}_{i \in \mathbf{A}} \right) = 1 \right] = 1.$$

Case 2: $\theta < |\mathbf{C}| < \Theta$, and \mathbf{C} is authorised. Let $|\mathbf{C}| = T$, such that $\theta < T < \Theta$ and \mathbf{C} is an authorised collection of players.

$$\text{Number of permutations that fix the access structure tokens of } \mathbf{C} = (\ell - T)!$$

$$\text{Total number of permutations on all } \ell \text{ access structure tokens} = \ell!$$

As there is a uniformly random distribution on the access structure tokens, \mathbf{C} can make a uniformly random guess from $\{0, 1\}$ about its authorisation status. Therefore, the probability that any collection of size T can identify itself as authorised can be bounded

above by the summation

$$\sum_{\substack{\mathbf{C} \in \Gamma \\ \text{with } |\mathbf{C}|=T}} \frac{(\ell - T)!}{\ell!} \leq \frac{1}{\binom{\ell}{T}},$$

and thus, $\Pr \left[\text{HsVer} \left(\left\{ \mathcal{U}_i^{(\Gamma)} \right\}_{i \in \mathbf{C}} \right) = 1 \right] \leq \sum_{\theta < T < \Theta} \frac{1}{\binom{\ell}{T}}.$ (5.3)

Denoting $\epsilon_1 := \sum_{\theta < T < \Theta} \frac{1}{\binom{\ell}{T}}$, we then have

$$\Pr \left[\text{HsVer} \left(\left\{ \mathcal{U}_i^{(\Gamma)} \right\}_{i \in \mathbf{C}} \right) = 1 \right] \geq 1 - \epsilon_1.$$

Soundness and ϵ_2 -soundness:

Case 1: $|\mathbf{B}| \leq \theta$. Since the access structure tokens of any collection of size at most θ never satisfy the logical condition (5.2), \mathbf{B} can simply check this condition and output 0. Therefore,

$$\Pr \left[\text{HsVer} \left(\left\{ \mathcal{U}_i^{(\Gamma)} \right\}_{i \in \mathbf{B}} \right) = 0 \right] = 1.$$

Case 2: $\theta < |\mathbf{C}| < \Theta$, and \mathbf{C} is unauthorised. Let $|\mathbf{C}| = T$, such that $\theta < T < \Theta$ and \mathbf{C} is an unauthorised collection of players. We arrive at the upper bound $\epsilon_2 := \sum_{\theta < T < \Theta} \frac{1}{\binom{\ell}{T}}$ as in Equation (5.3), by the same argument as for ϵ_1 -completeness above. Hence,

$$\Pr \left[\text{HsVer} \left(\left\{ \mathcal{U}_i^{(\Gamma)} \right\}_{i \in \mathbf{C}} \right) = 0 \right] \geq 1 - \epsilon_2.$$

Statistical hiding and ϵ_2 -statistical hiding: As $\mathcal{A} \otimes \mathcal{B}$ is a (θ, Θ, ℓ) -ramp scheme, any non-ramp collection of parties can simply count the access structure tokens of all its players and determine its authorisation.

Case 1: $|\mathbf{B}| \leq \theta$. By definition of the access structure tokens, $\bigvee_{i \in \mathbf{B}} \hat{\mathcal{U}}_i^{(1, \Gamma)} < \tau_1$ and $\bigvee_{i \in \mathbf{B}} \hat{\mathcal{U}}_i^{(2, \Gamma)} < \tau_2$.

Thus, for any such collection and for any access structure $\Gamma' \subseteq 2^{\mathbf{P}}$ characterised by the ramp bounds (θ, Θ) such that $\mathbf{B} \notin \Gamma'$, $\Gamma' \mid \left\{ \mathcal{U}_i^{(\Gamma')} \right\}_{i \in \mathbf{B}}$ follows the uniform distribution.

Hence,

$$\Pr \left[\Gamma' \mid \left\{ \mathcal{U}_i^{(\Gamma)} \right\}_{i \in \mathbf{B}} \right] = \frac{2}{\ell(\ell-3)} = \frac{2}{2^{b_1 b_2} (2^{b_1 b_2} - 3)}.$$

And therefore, $\left| \Pr \left[\Gamma \mid \left\{ \mathcal{U}_i^{(\Gamma)} \right\}_{i \in \mathbf{B}}, \left\{ s_i^{(k)} \right\}_{i \in \mathbf{B}} \right] - \Pr \left[\Gamma' \mid \left\{ \mathcal{U}_i^{(\Gamma)} \right\}_{i \in \mathbf{B}}, \left\{ s_i^{(k)} \right\}_{i \in \mathbf{B}} \right] \right| = 0$.

If Γ' is any other type of access structure (which does not characterise a ramp scheme), then $\Pr \left[\Gamma' \mid \left\{ \mathcal{U}_i^{(\Gamma)} \right\}_{i \in \mathbf{B}}, \left\{ s_i^{(k)} \right\}_{i \in \mathbf{B}} \right] = 0$.

And therefore, $\left| \Pr \left[\Gamma \mid \left\{ \mathcal{U}_i^{(\Gamma)} \right\}_{i \in \mathbf{B}}, \left\{ s_i^{(k)} \right\}_{i \in \mathbf{B}} \right] - \Pr \left[\Gamma' \mid \left\{ \mathcal{U}_i^{(\Gamma)} \right\}_{i \in \mathbf{B}}, \left\{ s_i^{(k)} \right\}_{i \in \mathbf{B}} \right] \right| = \frac{2}{2^{b_1 b_2} (2^{b_1 b_2} - 3)}$.

Case 2(a): $\theta < |\mathbf{C}| < \Theta$ and \mathbf{C} is unauthorised. Since \mathbf{C} is an unauthorised collection of parties, it knows no information about either factor, \mathcal{A} , \mathcal{B}_d , of $\mathcal{A} \otimes \mathcal{B}$. Therefore, by the same arguments as for Case 1,

$$\left| \Pr \left[\Gamma \mid \left\{ \mathcal{U}_i^{(\Gamma)} \right\}_{i \in \mathbf{B}}, \left\{ s_i^{(k)} \right\}_{i \in \mathbf{B}} \right] - \Pr \left[\Gamma' \mid \left\{ \mathcal{U}_i^{(\Gamma)} \right\}_{i \in \mathbf{B}}, \left\{ s_i^{(k)} \right\}_{i \in \mathbf{B}} \right] \right| = \frac{2}{2^{b_1 b_2} (2^{b_1 b_2} - 3)}.$$

Case 2(b): $\theta < |\mathbf{C}| < \Theta$ and \mathbf{C} has partial information about the secret. Let us assume \mathbf{C} knows the secret of the factor \mathcal{A} of $\mathcal{A} \otimes \mathcal{B}$. Then it must guess the shares of players of \mathcal{B}_d at best uniformly at random. So, a similar computation as in Case 1 allows us to arrive at the bound

$$\left| \Pr \left[\Gamma \mid \left\{ \mathcal{U}_i^{(\Gamma)} \right\}_{i \in \mathbf{B}}, \left\{ s_i^{(k)} \right\}_{i \in \mathbf{B}} \right] - \Pr \left[\Gamma' \mid \left\{ \mathcal{U}_i^{(\Gamma)} \right\}_{i \in \mathbf{B}}, \left\{ s_i^{(k)} \right\}_{i \in \mathbf{B}} \right] \right| \leq \frac{2}{2^{b_2} (2^{b_2} - 3)}.$$

On the other hand, if \mathbf{C} knows the secret of the factor \mathcal{B}_d of $\mathcal{A} \otimes \mathcal{B}$, then the bound becomes

$$\left| \Pr \left[\Gamma \mid \left\{ \mathcal{U}_i^{(\Gamma)} \right\}_{i \in \mathbf{B}}, \left\{ s_i^{(k)} \right\}_{i \in \mathbf{B}} \right] - \Pr \left[\Gamma' \mid \left\{ \mathcal{U}_i^{(\Gamma)} \right\}_{i \in \mathbf{B}}, \left\{ s_i^{(k)} \right\}_{i \in \mathbf{B}} \right] \right| \leq \frac{2}{2^{b_1} (2^{b_1} - 3)}.$$

The required value for the parameter ϵ_3 is therefore the maximum of these two bounds.

□

The proof of Theorem 5.4 is exactly similar to the proof above.

5.6 Proof of Theorems 5.5 and 5.6

Proof: If \mathbf{A} is an authorised collection of parties (irrespective of its size), then clearly,

$$\Pr \left[\mathbf{Ver} \left(k, \left\{ s_i^{(k)} \right\}_{i \in \mathbf{A}} \right) = 1 \right] = 1$$

as \mathbf{A} can reconstruct the secret perfectly.

Recall the definition of the prime power q from Section 5.3.1. For an unauthorised collection of parties \mathbf{A} such that \mathbf{A} cannot compute all elements of even one of \mathcal{A} or \mathcal{B}_d ,

$$\begin{aligned} \Pr \left[\mathbf{Ver} \left(k, \left\{ s_i^{(k)} \right\}_{i \in \mathbf{A}} \right) = 1 \right] &\leq \frac{1}{q} \\ \text{and therefore, } \Pr \left[\mathbf{Ver} \left(k, \left\{ s_i^{(k)} \right\}_{i \in \mathbf{A}} \right) = 0 \right] &\geq 1 - \frac{1}{q}. \end{aligned} \quad (5.4)$$

For a ramp collection of parties \mathbf{A} such that $\theta < |\mathbf{A}| < \Theta$, i.e. \mathbf{A} can compute all elements of exactly one of \mathcal{A} or \mathcal{B}_d ,

$$\begin{aligned} \Pr \left[\mathbf{Ver} \left(k, \left\{ s_i^{(k)} \right\}_{i \in \mathbf{A}} \right) = 1 \right] &\leq \max \left\{ \frac{1}{p_1}, \frac{1}{p_2} \right\} \\ \text{and therefore, } \Pr \left[\mathbf{Ver} \left(k, \left\{ s_i^{(k)} \right\}_{i \in \mathbf{A}} \right) = 0 \right] &\geq 1 - \max \left\{ \frac{1}{p_1}, \frac{1}{p_2} \right\}. \end{aligned} \quad (5.5)$$

The bounds in Equations (5.4) and (5.5) are simply because \mathcal{A} and \mathcal{B}_d are τ_1 - and τ_2 -threshold schemes based on Shamir schemes (Shamir, 1979), which means any collection of players that cannot reconstruct the entire secret cannot obtain any information about the secret. \square

The proof of Theorem 5.6 is exactly similar to the proof above.

5.7 Applications

Our technique has real-world applications in a very wide range of domains, including secure multiparty computation (Chaum, 1989; Andrychowicz, Dziembowski, Malinowski, & Mazurek, 2016; Smart, Baron, Saravanan, Brandt, & Mashatan, 2024), secure distributed storage (Garay, Gennaro, Jutla, & Rabin, 1997; Rajasekaran & Duraipandian, 2024), attribute-based encryption (Nali, Adams, & Miri, 2005; Ibraimi, Tang, Hartel, & Jonker, 2009; Saidi, Amira, &

Nouali, 2024; Asaithambi et al., 2024), access control mechanisms (Eland, 1978; di Vimercati, 2011; Gondara, 2011; Nour, Khelifi, Hussain, Mastorakis, & Moun gla, 2022), secure cloud computing (J. Xu, Huang, Huang, & Yang, 2009; Cui & Yi, 2024), e-voting systems (Rabia, Arezki, & Gadi, 2023), secure data sharing in blockchain technology (Zhang & Lin, 2018; Alshehri, Bamasag, Alghazzawi, & Jamjoom, 2023; Wang et al., 2023), and privacy-preserving machine learning algorithms (Çatak, 2015; K. Xu, Yue, Guo, Guo, & Fang, 2015; Qin et al., 2024; Mestari, Lenzini, & Demirci, 2024), to name a few.

For example in cloud storage systems (Shin, Koo, & Hur, 2017), our technique can enhance data integrity and availability by enabling authorized parties to reconstruct lost or corrupted shares without involving the initial dealer, avoiding framing of various parties, and computationally easy verification of shares against malicious adversary interactions.

Within sensor-based IoT systems (Sikder, Petracca, Aksu, Jaeger, & Uluagac, 2018), repairable ramp schemes safeguard the confidentiality and integrity of sensitive information exchanged among devices. The ability to repair lost or corrupted shares while maintaining frameproofness, and verifiability of these shares, along with the ability to ensure their completeness and soundness without the need to actually access the shares ensures uninterrupted operation and security, critical for IoT applications.

Furthermore, repairable ramp schemes are instrumental in multi-level security systems (Gao & Xiao, 2011; Wagner, 1997), such as those employed by government agencies and financial institutions. Our techniques would only improve their guarantees of security, while maintaining accessibility of critical information. They would also enable secure collaborative data sharing in environments where multiple parties require access to confidential data.

5.8 Conclusion and Future Work

In this chapter, we discuss verifiability and frameproofness of access structure hiding ramp-type tensor designs. We do this through the introduction of a new type of secret sharing scheme, called an ϵ -almost access structure hiding (θ, Θ, ℓ) -ramp tensor design, thus making an essential generalisation of the existing novel design introduced by Sehrawat et al.. We explore ways of enhancing data security and privacy, especially Roy et al.'s concept of extending repairable threshold schemes, using tensor products of balanced incomplete block designs. This concept

provides a fundamental generalization of existing designs, and thus plays an important role in enhancing the security and verifiability of secret sharing schemes by providing a mechanism for parties to verify the correctness of the shares they receive and ensuring that the reconstruction process is accurate. By incorporating ramp schemes, the construction becomes more robust against malicious behavior and unauthorized access, thus strengthening the overall security and integrity of the secret sharing process. We also list a few real-world applications where our techniques could be utilised for improved security.

6 | A Secret Sharing Application on a Public Transport Model

6.1 Introduction

We are now ready to formalize the ϵ -almost access structure hiding framework (A. Roy, Roy, Sakurai, & Talnikar, 2024) in a real-world context through a smart public transportation system running buses through a country. The main goal of this set-up will be to protect the private data and travel history of passengers using the bus service while securely and efficiently running the system. Various entities such as the passengers themselves, the government, different bus companies, etc. are a part of the system, each with a different status in a hierarchical structure. We shall model this example by introducing a new ramp-type hierarchical secret sharing scheme motivated by (Tassa, 2007). We shall also incorporate the ledger update protocol of (Dutta et al., 2021) within this framework. We shall discuss applications of this scheme in IoT as well as other use-cases such as in ledger management situations. Finally, we shall also describe a verification protocol through a good lightweight authenticated encryption scheme, say ASCON (Dobraunig et al., 2021).

6.1.1 An Overview of the Model

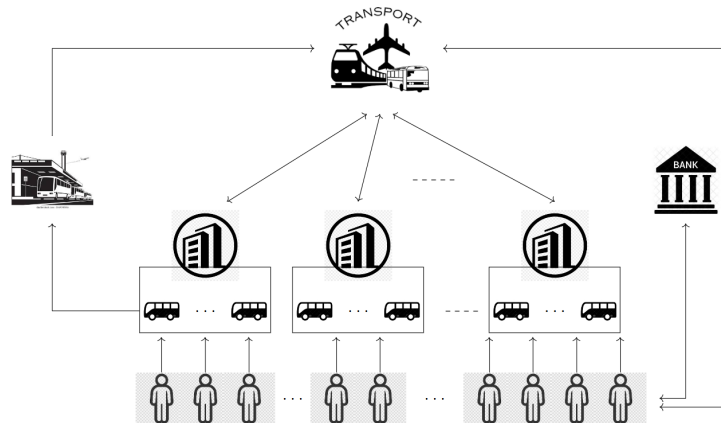


Figure 6.1: An overview of the communication flow in the transport network.

The main goal of our secret sharing model in the context of this public transportation system is to protect the private data and travel history of passengers while securely and efficiently running

the transportation system. This involves ensuring that various entities, such as passengers, the government, and different bus companies, can operate within a hierarchical structure without compromising sensitive information. We assume that every passenger has a travel id card and a bank account. Let us recall the questions from Chapter 1, which we shall soon answer with our framework:

- Each bus maintains a ledger. With whom does it communicate this?
- How to encrypt this bus ledger?
- How to consolidate it with the bus station ledger?
- What can be a good ledger update protocol for \mathcal{E} ?
- Who can read the ledger(s)?
- In which communication channels can errors occur?
- Which communication channels can be affected by malicious entities?
- Which participants can be affected by framing attacks from other participants?
- How to ensure secure communication (verifiability)?
- How to protect the participants from framing and other attacks?
- How to ensure that our system is lightweight, secures all passenger data, and satisfies all the above requirements?

Modelling the Scheme

We design a ramp-type hierarchical secret sharing model that allows for flexible secret reconstruction, using Birkhoff's interpolation to distribute smaller secrets amongst designated players. We assign the highest priority to the Transport Department and second highest to the Bus Companies running buses (which have the lowest priority). Other entities such as Passengers, the Station Ledger and the Bank are not assigned any priority as they do not contribute any shares. The smaller secret sharing schemes are viewed together as a single tensor design. Several verification protocols like secure lightweight encryption and secure ledger updation ensure that the scheme is sufficiently lightweight to be feasible for such a multi-level model, and all communication

channels exchange correct information. Section 6.2 provides a thorough explanation of this implementation.

6.1.2 Communication Flow

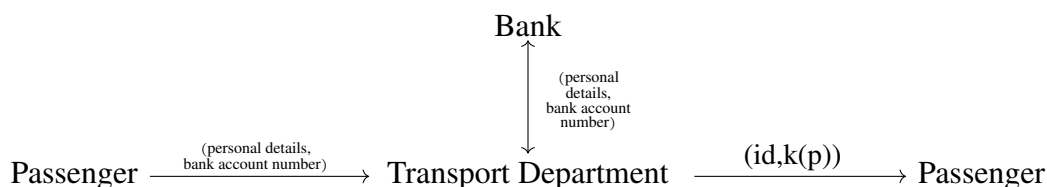
There is an interconnected network of communication between the various entities involved in this model. Passengers are individual people who use the bus transportation service, the Station Ledger is a record of all bus stops and the buses that arrive at these stops throughout the day, various Bus Companies manage a number of Buses running on different routes every day, the Transportation Department is a government entity that runs the bus transportation system, and the bank allows the Transportation Department to deduct all travel charges incurred by each passenger. A high-level visualisation of the communication flow is shown in figure 6.1.

Station Ledger: Suppose there are M bus stations, numbered $\{m\}_{m \in \{1, \dots, M\}}$. Also let the buses be numbered $1, 2, \dots, R$, where $R = r_1 + r_2 + \dots + r_n$, and the i^{th} bus company C_i controls buses numbered $r_{i-1} + 1, r_{i-1} + 2, \dots, r_{i-1} + r_i$. Finally, assume there to be a total of T timestamps (say $T = 1440 = 24 \text{ hours} \times 60 \text{ minutes}$) in a single running day. Then the bus station ledger is a collection of all data of the form

$$\{(m, r, t)\},$$

where $m \in \{1, 2, \dots, M\}$ is the bus station at which bus b_r ($r \in \{1, 2, \dots, R\}$) stops at time $t \in \{1, 2, \dots, T\}$.

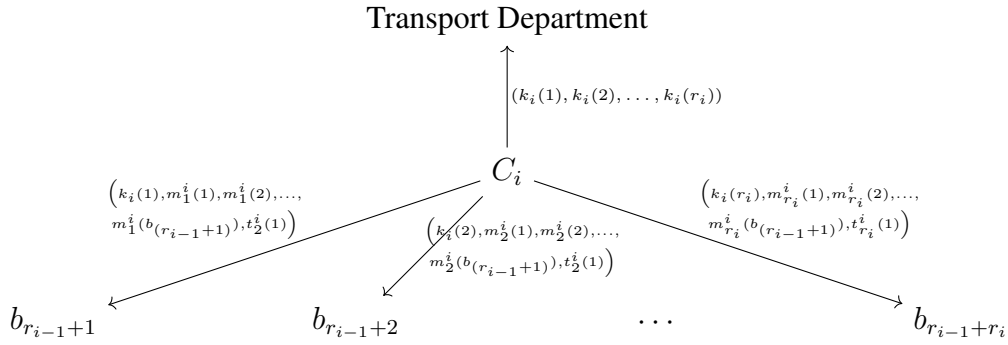
Transportation Department to Passenger: Each (potential) passenger applies to the transportation department for a travel card. The transportation department (physically) sends such a card (with a unique passenger identity id and a private encryption key $k(p)$) to the applicant after linking it with his/her bank account.



Bus Company to Bus: By ‘one bus’, we mean a bus running on exactly one route exactly

once a day. Therefore, if the same (physical) bus runs two different routes in a day, it is considered as two different buses in our bus count. First of all, each bus b_r receives a secret key $k_b(r)$ from its respective company at the beginning of each day (possibly through a standard public key exchange protocol, which we shall not discuss here). Additionally, these keys are also communicated with the Transport Department.

At the beginning of each day (i.e. at $t := 0$), each bus receives the route it must follow at an ordered sequence of bus station numbers, along with starting time. Thus, each bus b receives $(m_1, m_2, \dots, m_b, t^b(1))$ from its respective company C .



Bus to Bus Company: At the end of the running day, each bus b_r under company C_i uses its key $k_b(r)$ (and the day as IV) to encrypt through ASCON its actual route of the day, and sends it to C_i . It similarly encrypts its passenger travel records, and also sends it to C_i .

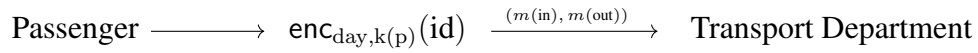
$$\text{Bus } b_{r_{i-1}+r} \xrightarrow{\frac{\text{enc}_{\text{day}, k_b(r)}(\bar{m}_r^i(1), \bar{m}_r^i(2), \dots, \bar{m}_r^i(b_r))}{(\text{enc}(\text{id}), m(\text{in}), m(\text{out}))}} \text{Company } C_i$$

Passenger to Bus: Each passenger scans their card when getting on (check-in) and off (check-out) a bus. While checking in, the passenger's encrypted id along with the bus station where he/she boarded the bus (i.e. $(\text{enc}(\text{id}), m(\text{in}))$) is recorded. This record is updated with the bus station when the passenger alights the bus

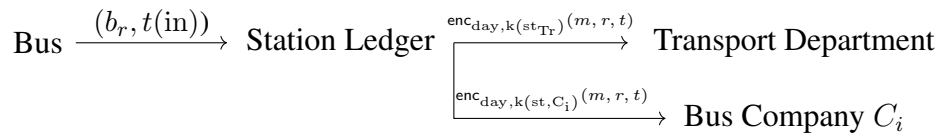
$$\text{i.e. } (\text{enc}(\text{id}), m(\text{in}, m(\text{out}))).$$

Furthermore, each passenger id is encrypted through a good lightweight block cipher using the day of travel as the IV/nonce/counter/tweak and his/her private key (shared by the Travel Department with the passenger when physically sending the travel card). (D. Sehwat

& Gill, 2018) presents a comparative study of various lightweight block ciphers suitable for IoT applications along with their benefits and limitations. Given a Passenger p , we call this secret key $k(p)$. Due to this physical exchange of information, we assume a secure secret key exchange between the Transport Department and the Passenger.

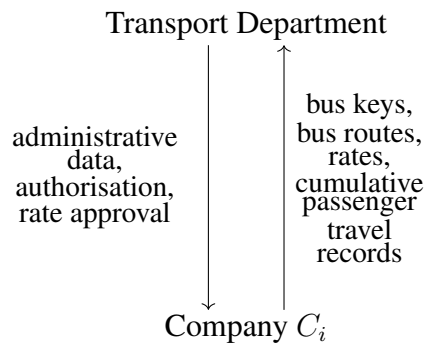


Bus to Station Ledger: Each bus sends its identification number (i.e. r) and its time of arrival to each bus station on its route. The bus station records this information along with its own identification number (i.e. m) in the common Station Ledger maintained for all bus stations. This ledger is encrypted using the day as the IV and some keys $k(\text{st}_{\text{Tr}})$ and $k(\text{st}, C_i)$ through ASCON, and sent to the Transportation Department as well as the company (say C_i) of bus b_r , respectively, at the end of each running day.

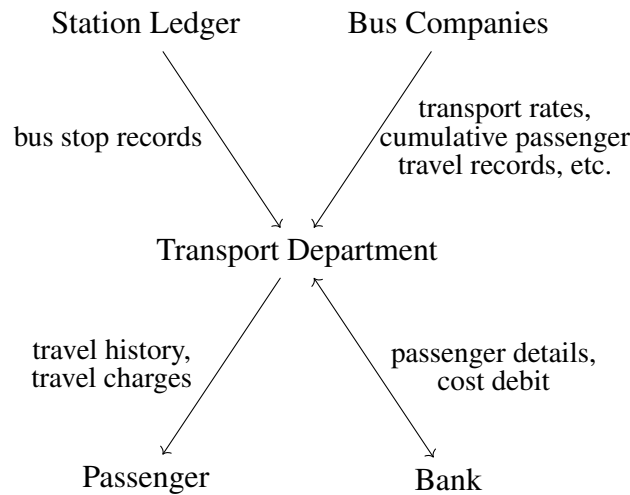


Bus Company to Transport Department: The Transport Department sends administrative data such as authorisation to run buses on various routes to each bus company.

Once this is done, the information communicated by each bus company with the Transport Department includes the secret key $k_b(r)$ that each bus b_r receives from it at the beginning of each day, the transport rates for all the routes run by its buses (along with the actual bus routes), and (possibly) the cumulative passenger travel records from all its buses at the end of a day.



Transport Department to Bank and Passenger: Finally, the Transport Department consolidates all travel information of each passenger; at the end of a certain time period (say, one month), it directly debits the travel costs from the connected bank account of each passenger. It also allows a view of the passenger’s travel history and charges incurred on a secure platform, only to the passenger.



Malicious Entities

Next, we enumerate which entities in the network can be malicious. Since the system is formed (and run) by the government, we assume all government entities, i.e. the station ledger (and therefore also all bus stops) as well as the transport department as non-malicious. If all bus companies are malicious, it is not possible to run the system. Therefore, we also assume that a majority of the companies are honest. A number of (including the case when all) buses – under a single company or even under all companies – may be malicious; any false data computed by the corresponding company will be captured by the transport department. For this verification, the transport department runs a ledger updation algorithm inspired by the one from (Dutta et al., 2021). The transport department can similarly take note of only a small number of malicious bus companies, given that all other bus companies communicate honestly and without error.

As is the case for buses, so too can all passengers be malicious. Passenger maliciousness may occur in two ways: (i) *Incorrect feedback to the transport department*. This must simply be controlled by the complaint resolution policy of the transport department. (ii) *Collusion with a bus*. A bus may collect money personally (say by cash) and allow passengers to travel

without scanning their travel cards. This can be resolved by installing a camera (or say, an infra-red scanner) on all buses, which would count the total number of passengers boarding and deboarding; the bus ledger is appended with this counter at the end of the day before being sent to the respective bus company. Finally an adversary (i.e. an outside malicious entity) may attempt to attack any part of the entire system, and the system is protected from such attacks by the nature of its secret sharing model.

Framing Attacks

All passengers are protected by the secret key sent physically with the travel card from the transport department, through a good lightweight block cipher. Any collection of buses being framed by a collusion of buses is protected as the secret sharing scheme is frameproof. Likewise any collection of bus companies being framed by a collusion of other bus companies is also protected as the secret sharing scheme (which shall be the tensor design of chapter 2) is frameproof. If a collusion of buses tries to frame a bus company, the error will be captured by the transport department similarly as in the case of malicious buses.

Errors in the Communication Network

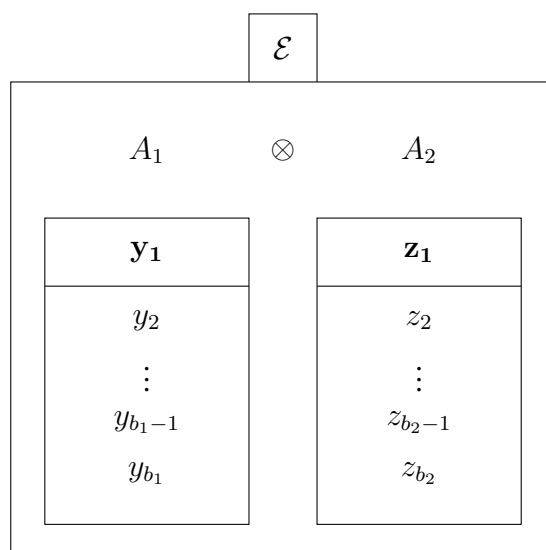
A bus may get a wrong key from its company at the start of a running day. The bus encrypts its log using this key, and sends this encrypted log to the bus company at the end of the day. Due to the ledger updation protocol of the transport department, the bus company becomes aware that the information provided by the bus is incorrect; it (actually, the transport department) may even be able to compute that the error is in the encryption key of the bus ledger. The bus company can retrieve this wrong key so as to decrypt the bus log by simply asking the particular bus for the key; this can be achieved by ensuring that every bus has memory space to store two (or three) daily keys at any given time, so that the company can request old keys up to two days previously from a bus. Similarly, the transport department can also compute when it receives incorrect information from any bus company (due to its ledger updation protocol), and can notify it to resend its information. The transport department can also verify the information it receives from the station ledger and bus companies by comparing the two logs and searching for mismatches.

We assume that there is no error in certain communication channels, such as from the passenger to the bus (any passenger whose travel card does not scan correctly is not allowed to travel in a

bus), all buses can successfully record their arrival at all bus stops, and the transport department can also communicate with the bank(s) and passengers without any errors.

6.2 Implementation

Our objective is to secure passengers' private travel data and payment log from various malicious entities (such as the companies). For this, we must incorporate access structure tokens and (ϵ -almost) access structure hiding into a suitable verifiable and frameproof combinatorial secret sharing scheme. We must also clearly enumerate which entities can be malicious parties (points of error in the communication network) and formulate secure and efficient verification/error correction algorithm(s) to be used in the various transactions/communications in this scheme. We propose the lightweight authenticated encryption scheme ASCON to be used for verification purposes in our model.



Let us first consider the base case of only two secret sharing schemes, A_1 and A_2 . We propose a hierarchical secret sharing structure through Birkhoff's interpolation to distribute the smaller secrets s_1 and s_2 amongst the players $\{y_1, y_2, \dots, y_{b_1}\}$ and $\{z_1, z_2, \dots, z_{b_2}\}$, respectively. Ramp bounds are maintained as the authorised sets are derived from $A_1 \otimes A_2$. y_1 and z_1 are the bus companies; all other (non-priority) players are the buses. The goal of the secret sharing scheme is to secure an individual passenger's travel data from (most of the) other players, and to regularise the bus rates and other information of each bus company with the transport department of the government (which is \mathcal{E} in this case).

Ramp Bounds

Recall the setup of a ramp scheme with ramp bounds θ and Θ as defined in 5.1 of Chapter 5. Since this chapter proposes implementing a ramp-type scheme, the same notation is carried over here.

- \mathcal{E} views the smaller schemes altogether as a single tensor design.
- Thus, $\theta = (\tau_1 - 1)(\tau_2 - 1) + 1$, where the priority share $\mathbf{y}_1\mathbf{z}_1$, any $\tau_1 - 1$ of the shares \mathbf{z}_1y_i , $i \in \{2, \dots, b_1\}$, any $\tau_2 - 1$ of the shares \mathbf{y}_1z_j , $j \in \{2, \dots, b_2\}$, and one more from any of the remaining shares y_iz_j , $i \in \{1, \dots, b_1\}$, $j \in \{1, \dots, b_2\}$ are enough to reconstruct the secret s .
- $\Theta = \min \{(\tau_1 - 1)b_2 + 1, (\tau_2 - 1)b_1 + 1\}$, since the priority share $\mathbf{y}_1\mathbf{z}_1$ along with the remaining number of any of the remaining shares y_iz_j , $i \in \{1, \dots, b_1\}$, $j \in \{1, \dots, b_2\}$ can always reconstruct the secret s .

6.3 Conclusion

This chapter consolidates the theoretical underpinnings established throughout this thesis by introducing a novel secret sharing model tailored to safeguard the privacy of passenger data and travel histories within a smart public transportation system while ensuring the framework's secure and efficient operation. The primary objective of this secret sharing framework is to protect sensitive passenger information and travel records while maintaining the system's functionality. This necessitates a hierarchical structure that enables diverse entities to collaborate without compromising data confidentiality. The proposed framework establishes a secure mechanism for data sharing and verification among these entities, thereby enhancing the overall security and integrity of the public transportation system.

Building upon the foundation laid by Verheul et al., Feldman, Pedersen, Stinson, Desmedt et al., Sehrawat et al., and Roy et al.'s seminal contributions to verifiable secret sharing, frameproofness, and tensor-based designs, we introduce a ramp-type hierarchical secret sharing scheme. This approach leverages the concept of hierarchical secret sharing by assigning varying levels of access to different entities (passengers, government, bus companies) while employing tensor design to bolster security and streamline secret reconstruction. To mitigate potential

vulnerabilities such as communication errors, framing attacks, and malicious behavior, the framework incorporates lightweight block ciphers, authenticated encryption, and the inherent frameproofness and verifiability of the underlying tensor design.

To concretize the ϵ -almost access structure hiding framework in a real-world context, we present a smart public transportation system comprising passengers, government, bus companies, and a bank. The system's primary goal is to protect passenger privacy while ensuring efficient operations. Our proposed ramp-type hierarchical secret sharing scheme, inspired by Tassa's work, assigns distinct priorities to participants and utilizes Birkhoff's interpolation for flexible secret reconstruction. The interconnected network encompasses passengers, station ledgers, bus companies, the transportation department, and a bank, each with specific roles and potential vulnerabilities. By integrating the ledger updation protocol of Dutta et al. and employing a robust lightweight authenticated encryption scheme like Ascon, we provide a comprehensive cryptographic implementation that safeguards data integrity and privacy.

In summary, this chapter offers a novel and practical approach to enhancing the security and efficiency of public transportation systems through a carefully designed secret sharing framework. By addressing the challenges posed by data privacy, hierarchical access control, and potential threats, this work contributes significantly to the advancement of secure and reliable public transportation solutions.

7 | Conclusions and Open Issues

In this thesis, we have first generalized the concept of combinatorial RTS and then improved our secret sharing scheme by producing a frameproof one. We believe our results can be extended further to an arbitrary number of distribution designs. We also believe that the Krönecker product of BIBDs can be generalized to t -designs, and all corresponding results will hold for these. Furthermore, we have discussed the extensive scope of applicability for our proposed scheme in a diverse array of IoT contexts. A fascinating avenue for further investigation entails the examination of specific instances of these applications.

We have also shown an efficient method of verification through a cheater identification algorithm that makes our construction a verifiable secret sharing scheme and greatly improves its suitability for various IoT applications. While we have reduced the storage space requirements, the size of communication increases. Decreasing this is an interesting future problem to consider. Additionally, conducting rigorous experimental evaluations to validate our theoretical findings and assess the scheme's performance under real-world conditions is a promising research direction.

We have discussed verifiability and frameproofness of access structure hiding ramp-type tensor designs through the introduction of a new type of secret sharing scheme, called an ϵ -almost access structure hiding tensor design, thus making an essential generalisation of the existing novel design introduced by Serawat et al.. We have explored ways of enhancing data security and privacy, especially Roy et al.'s concept of extending repairable threshold schemes, using tensor products of balanced incomplete block designs. This concept provides a fundamental generalization of existing designs, and thus plays an important role in enhancing the security and verifiability of secret sharing schemes by providing a mechanism for parties to verify the correctness of the shares they receive and ensuring that the reconstruction process is accurate. By incorporating ramp schemes, the construction becomes more robust against malicious behavior and unauthorized access, thus strengthening the overall security and integrity of the secret sharing process. We have also listed a few real-world applications where our techniques could be utilised for improved security.

While we demonstrate our concept of ϵ -almost access structure hiding for only extendable combinatorial tensor designs, it opens up a wide range of possibilities for any ramp-type scheme

to incorporate this technique for further improvement of confidentiality, secrecy and verifiability.

Finally, we have implemented our abstract construction of a tensor design into a real-life model, where we have shown that even in the case of small storage and fast computability requirements, sufficient security can be provided through a consolidation of all our theoretical concepts.

To fully realize the potential of our proposed scheme, future research should focus on several key areas. Extending the framework to support more complex access structures and dynamic group management is essential. Conducting comprehensive performance evaluations and security analyses under various threat models is crucial for assessing the scheme's practical viability. Finally, developing efficient implementations for different hardware platforms can realise the scheme's applicability in resource-constrained environments.

By addressing these research directions, we can further advance the state-of-the-art in secret sharing and unlock new opportunities for secure and privacy-preserving applications.

Bibliography

- AES. (2001). *Advanced Encryption standard (AES)*. (Vols. (Department of Commerce, Washington, D.C.), Federal Information Processing Standards Publication (FIPS) NIST FIPS 197-upd1, updated May 9, 2023). Retrieved from <https://doi.org/10.6028/NIST.FIPS.197-upd1>
- Alshehri, S., Bamasag, O., Alghazzawi, D. M., & Jamjoom, A. (2023). Dynamic Secure Access Control and Data Sharing Through Trusted Delegation and Revocation in a Blockchain-Enabled Cloud-IoT Environment. *IEEE Internet Things J.*, *10*(5), 4239–4256. Retrieved from <https://doi.org/10.1109/JIOT.2022.3217087> doi: 10.1109/JIOT.2022.3217087
- Andrychowicz, M., Dziembowski, S., Malinowski, D., & Mazurek, L. (2016). Secure Multiparty Computations on Bitcoin. *Commun. ACM*, *59*(4), 76–84. Retrieved from <https://doi.org/10.1145/2896386> doi: 10.1145/2896386
- Asaithambi, S., Ravi, L., Devarajan, M., Selvalakshmi, A., Almaktoom, A. T., Almazyad, A. S., ... Mohamed, A. W. (2024). Blockchain-Assisted Hierarchical Attribute-Based Encryption Scheme for Secure Information Sharing in Industrial Internet of Things. *IEEE Access*, *12*, 12586–12601. Retrieved from <https://doi.org/10.1109/ACCESS.2024.3354846> doi: 10.1109/ACCESS.2024.3354846
- Benaloh, J. C. (1986). Secret Sharing Homomorphisms: Keeping Shares of A Secret Sharing. In A. M. Odlyzko (Ed.), *Advances in cryptology - CRYPTO '86, santa barbara, california, usa, 1986, proceedings* (Vol. 263, pp. 251–260). Springer. Retrieved from https://doi.org/10.1007/3-540-47721-7_19 doi: 10.1007/3-540-47721-7_19
- Cafaro, M., & Pellè, P. (2014). Space-Efficient Non-Interactive Verification Enhancing Arbitrary Secret Sharing Schemes by Adding Cheater Detection Capabilities. *ArXiv, abs/1401.7471*. Retrieved from <https://api.semanticscholar.org/CorpusID:195347263>
- Cafaro, M., & Pellè, P. (2018). Space-Efficient Verifiable Secret Sharing Using Polynomial Interpolation. *IEEE Trans. Cloud Comput.*, *6*(2), 453–463. Retrieved from <https://doi.org/10.1109/TCC.2015.2396072> doi: 10.1109/TCC.2015.2396072
- Çatak, F. Ö. (2015). Secure Multi-party Computation Based Privacy Preserving Extreme Learning Machine Algorithm over Vertically Distributed Data. In S. Arik, T. Huang,

- W. K. Lai, & Q. Liu (Eds.), *Neural information processing - 22nd international conference, ICONIP 2015, istanbul, turkey, november 9-12, 2015, proceedings, part II* (Vol. 9490, pp. 337–345). Springer. Retrieved from https://doi.org/10.1007/978-3-319-26535-3_39 doi: 10.1007/978-3-319-26535-3_39
- Chaum, D. (1989). The Spymasters Double-Agent Problem: Multiparty Computations Secure Unconditionally from Minorities and Cryptographically from Majorities. In G. Brassard (Ed.), *Advances in cryptology - CRYPTO '89, 9th annual international cryptology conference, santa barbara, california, usa, august 20-24, 1989, proceedings* (Vol. 435, pp. 591–602). Springer. Retrieved from https://doi.org/10.1007/0-387-34805-0_52 doi: 10.1007/0-387-34805-0_52
- Cui, H., & Yi, X. (2024). Secure Internet of Things in Cloud Computing via Puncturable Attribute-Based Encryption with User Revocation. *IEEE Internet Things J.*, 11(2), 3662–3670. Retrieved from <https://doi.org/10.1109/JIOT.2023.3297997> doi: 10.1109/JIOT.2023.3297997
- De Cannière, C. (2005). Triple-DES. In H. C. A. van Tilborg (Ed.), *Encyclopedia of cryptography and security* (pp. 626–627). Boston, MA: Springer US. Retrieved from https://doi.org/10.1007/0-387-23483-7_437 doi: 10.1007/0-387-23483-7_437
- Dehkordi, M. H., Farahi, S. T., & Mashhadi, S. (2024). LWE-based Verifiable Essential Secret Image Sharing Scheme $((t , s , k , n) \$(\{ t,s,k,n \})\$ - VESIS). *IET Image Process.*, 18(4), 1053–1072. Retrieved from <https://doi.org/10.1049/ipr2.13006> doi: 10.1049/IPR2.13006$
- DES. (1979). *FIPS-46: Data Encryption Standard (DES)* (Vols. Revised as FIPS 46-1:1988, FIPS 46-2:1993, FIPS 46-3:1999). Retrieved from <http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf>
- Desmedt, Y., Mo, S., & Slinko, A. M. (2021). Framing in Secret Sharing. *IEEE Trans. Inf. Forensics Secur.*, 16, 2836–2842. Retrieved from <https://doi.org/10.1109/TIFS.2021.3067468> doi: 10.1109/TIFS.2021.3067468
- di Vimercati, S. D. C. (2011). Access Control Policies, Models, and Mechanisms. In H. C. A. van Tilborg & S. Jajodia (Eds.), *Encyclopedia of cryptography and security, 2nd ed* (pp. 13–14). Springer. Retrieved from https://doi.org/10.1007/978-1-4419-5906-5_806 doi: 10.1007/978-1-4419-5906-5_806
- Dobraunig, C., Eichlseder, M., Mendel, F., & Schl affer, M. (2021). Ascon v1.2: Lightweight

- Authenticated Encryption and Hashing. *J. Cryptol.*, 34(3), 33. Retrieved from <https://doi.org/10.1007/s00145-021-09398-9> doi: 10.1007/S00145-021-09398-9
- Dong, D., Mani, N., & Zhao, Y. (2023). On the Number of Error Correcting Codes. *Comb. Probab. Comput.*, 32(5), 819–832. Retrieved from <https://doi.org/10.1017/s0963548323000111> doi: 10.1017/S0963548323000111
- Dutta, S., Paul, A., Ozaki, R. H., Ranzan, C. P., & Sakurai, K. (2021). A Distributed Ledger Management Mechanism for Storing and Selling Private Data. In *2021 IEEE Conference on Dependable and Secure Computing (DSC)* (p. 1-8). doi: 10.1109/DSC49826.2021.9346258
- Eland, N. (1978). *Language-Based Access Control Mechanisms for Shared Databases* (Unpublished doctoral dissertation). Cornell University, USA.
- Feldman, P. (1987). A Practical Scheme for Non-interactive Verifiable Secret Sharing. In *28th annual symposium on foundations of computer science, los angeles, california, usa, 27-29 october 1987* (pp. 427–437). IEEE Computer Society. Retrieved from <https://doi.org/10.1109/SFCS.1987.4> doi: 10.1109/SFCS.1987.4
- Fu, Y., Ren, Y., Feng, G., Zhang, X., & Qin, C. (2021). Non-Interactive and Secure Data Aggregation Scheme for Internet of Things. *Electronics*, 10(20). Retrieved from <https://www.mdpi.com/2079-9292/10/20/2464> doi: 10.3390/electronics10202464
- Gao, C., & Xiao, C. (2011). A Security Model for Information Systems with Multi-level Security. In Y. Wang, Y. Cheung, P. Guo, & Y. Wei (Eds.), *Seventh international conference on computational intelligence and security, CIS 2011, sanya, hainan, china, december 3-4, 2011* (pp. 620–624). IEEE Computer Society. Retrieved from <https://doi.org/10.1109/CIS.2011.142> doi: 10.1109/CIS.2011.142
- Garay, J. A., Gennaro, R., Jutla, C. S., & Rabin, T. (1997). Secure Distributed Storage and Retrieval. In M. Mavronicolas & P. Tsigas (Eds.), *Distributed algorithms, 11th international workshop, WDAG '97, saarbrücken, germany, september 24-26, 1997, proceedings* (Vol. 1320, pp. 275–289). Springer. Retrieved from <https://doi.org/10.1007/BFb0030690> doi: 10.1007/BFB0030690
- Geng, T., Njilla, L., & Huang, C. (2022). Delegated Proof of Secret Sharing: A Privacy-Preserving Consensus Protocol Based on Secure Multiparty Computation for IoT Environment. *Network*, 2(1), 66–80. Retrieved from <https://doi.org/10.3390/network2010005> doi: 10.3390/NETWORK2010005

- Gondara, M. K. (2011). Access Control Mechanisms for Semantic Web Services-A Discussion on Requirements & Future Directions. *CoRR*, *abs/1105.0141*. Retrieved from <http://arxiv.org/abs/1105.0141>
- Hall, P. (1935, 01). On Representatives of Subsets. *Journal of the London Mathematical Society*, *s1-10*(1), 26-30. Retrieved from <https://doi.org/10.1112/jlms/s1-10.37.26> doi: 10.1112/jlms/s1-10.37.26
- Harn, L., & Lin, C. (2009). Detection and Identification of Cheaters in (t, n) Secret Sharing Scheme. *Des. Codes Cryptogr.*, *52*(1), 15–24. Retrieved from <https://doi.org/10.1007/s10623-008-9265-8> doi: 10.1007/S10623-008-9265-8
- Ibraimi, L., Tang, Q., Hartel, P. H., & Jonker, W. (2009). Efficient and Provable Secure Ciphertext-Policy Attribute-Based Encryption Schemes. In F. Bao, H. Li, & G. Wang (Eds.), *Information security practice and experience, 5th international conference, ISPEC 2009, xi'an, china, april 13-15, 2009, proceedings* (Vol. 5451, pp. 1–12). Springer. Retrieved from https://doi.org/10.1007/978-3-642-00843-6_1 doi: 10.1007/978-3-642-00843-6_1
- Kacsmar, B., & Stinson, D. R. (2019). A Network Reliability Approach to the Analysis of Combinatorial Repairable Threshold Schemes. *Adv. Math. Commun.*, *13*(4), 601–612. Retrieved from <https://doi.org/10.3934/amc.2019037> doi: 10.3934/AMC.2019037
- König, D. (1931). *Gráfok és Mátrixok*. Matematikai és Fizikai Lapok.
- Laing, T. M., & Stinson, D. R. (2018). A Survey and Refinement of Repairable Threshold Schemes. *J. Math. Cryptol.*, *12*(1), 57–81. Retrieved from <https://doi.org/10.1515/jmc-2017-0058> doi: 10.1515/JMC-2017-0058
- Luo, Y., Deng, X., Wu, Y., & Wang, J. (2019). Mpc-dpos: An Efficient Consensus Algorithm Based on Secure Multi-Party Computation. In *Proceedings of the 2019 2nd international conference on blockchain technology and applications* (pp. 105–112).
- Mestari, S. Z. E., Lenzini, G., & Demirci, H. (2024). Preserving Data Privacy in Machine Learning Systems. *Comput. Secur.*, *137*, 103605. Retrieved from <https://doi.org/10.1016/j.cose.2023.103605> doi: 10.1016/J.COSE.2023.103605
- Nali, D., Adams, C. M., & Miri, A. (2005). Using Threshold Attribute-based Encryption for Practical Biometric-based Access Control. *Int. J. Netw. Secur.*, *1*(3), 173–182. Retrieved from <http://ijns.jalaxy.com.tw/contents/ijns-v1-n3/ijns-2005>

-v1-n3-p173-182.pdf

- Nirmala, S. J., Bhanu, S. M. S., Patel, A. A., & Pvt, O. I. (2012). A Comparative Study of the Secret Sharing Algorithms for Secure Data in the Cloud. In *International conference on cloud computing*. Retrieved from <https://api.semanticscholar.org/CorpusID:50283858>
- Nour, B., Khelifi, H., Hussain, R., Mastorakis, S., & Mounsla, H. (2022). Access Control Mechanisms in Named Data Networks: A Comprehensive Survey. *ACM Comput. Surv.*, 54(3), 61:1–61:35. Retrieved from <https://doi.org/10.1145/3442150> doi: 10.1145/3442150
- Park, N., & Lee, D. (2018). Electronic Identity Information Hiding Methods Using a Secret Sharing Scheme in Multimedia-Centric Internet of Things Environment. *Pers. Ubiquitous Comput.*, 22(1), 3–10. Retrieved from <https://doi.org/10.1007/s00779-017-1017-1> doi: 10.1007/S00779-017-1017-1
- Paterson, M. B., & Stinson, D. R. (2013). A Simple Combinatorial Treatment of Constructions and Threshold Gaps of Ramp Schemes. *Cryptogr. Commun.*, 5(4), 229–240. Retrieved from <https://doi.org/10.1007/s12095-013-0082-1> doi: 10.1007/S12095-013-0082-1
- Pedersen, T. P. (1991). Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing. In J. Feigenbaum (Ed.), *Advances in cryptology - CRYPTO '91, 11th annual international cryptology conference, santa barbara, california, usa, august 11-15, 1991, proceedings* (Vol. 576, pp. 129–140). Springer. Retrieved from https://doi.org/10.1007/3-540-46766-1_9 doi: 10.1007/3-540-46766-1_9
- Peng, K. (2012). Critical Survey of Existing Publicly Verifiable Secret Sharing Schemes. *IET Inf. Secur.*, 6(4), 249–257. Retrieved from <https://doi.org/10.1049/iet-ifs.2011.0201> doi: 10.1049/IET-IFS.2011.0201
- Poli, A. (1985). Some Algebraic Tools for Error-Correcting Codes. In J. Calmet (Ed.), *Algebraic algorithms and error-correcting codes, 3rd international conference, aaecc-3, grenoble, france, july 15-19, 1985, proceedings* (Vol. 229, pp. 43–60). Springer. Retrieved from https://doi.org/10.1007/3-540-16776-5_708 doi: 10.1007/3-540-16776-5_708
- Qin, H., He, D., Feng, Q., Khan, M. K., Luo, M., & Choo, K. R. (2024). Cryptographic Primitives in Privacy-Preserving Machine Learning: A Survey. *IEEE Trans. Knowl.*

- Data Eng.*, 36(5), 1919–1934. Retrieved from <https://doi.org/10.1109/TKDE.2023.3321803> doi: 10.1109/TKDE.2023.3321803
- Rabia, F., Arezki, S., & Gadi, T. (2023). A Review of Blockchain-Based e-Voting Systems: Comparative Analysis and Findings. *Int. J. Interact. Mob. Technol.*, 17(23), 49–67. Retrieved from <https://doi.org/10.3991/ijim.v17i23.45257> doi: 10.3991/IJIM.V17I23.45257
- Rabin, M. O. (1989, apr). Efficient Dispersal of Information for Security, Load Balancing, and Fault Tolerance. *J. ACM*, 36(2), 335-348. Retrieved from <https://doi.org/10.1145/62044.62050> doi: 10.1145/62044.62050
- Rajasekaran, P., & Duraipandian, M. (2024). Secure Cloud Storage for IoT Based Distributed Healthcare Environment Using Blockchain Orchestrated and Deep Learning Model. *J. Intell. Fuzzy Syst.*, 46(1), 1069–1084. Retrieved from <https://doi.org/10.3233/jifs-234884> doi: 10.3233/JIFS-234884
- Rehman, A., Saba, T., Haseeb, K., Larabi Marie-Sainte, S., & Lloret, J. (2021). Energy-Efficient IoT e-Health Using Artificial Intelligence Model with Homomorphic Secret Sharing. *Energies*, 14(19). Retrieved from <https://www.mdpi.com/1996-1073/14/19/6414> doi: 10.3390/en14196414
- Rivest, R. L. (1994). The RC5 Encryption Algorithm. In B. Preneel (Ed.), *Fast software encryption: Second international workshop. leuven, belgium, 14-16 december 1994, proceedings* (Vol. 1008, pp. 86–96). Springer. Retrieved from https://doi.org/10.1007/3-540-60590-8_7 doi: 10.1007/3-540-60590-8_7
- Roy, A., Roy, B. K., Sakurai, K., & Talnikar, S. (2024). Access Structure Hiding Verifiable Tensor Designs. *IACR Cryptol. ePrint Arch.*, 902. Retrieved from <https://eprint.iacr.org/2024/902>
- Roy, B. K., & Roy, A. (2023). IoT-Applicable Generalized Frameproof Combinatorial Designs. *IoT*, 4(3), 466–485. Retrieved from <https://www.mdpi.com/2624-831X/4/3/20> doi: 10.3390/iot4030020
- Saidi, A., Amira, A., & Nouali, O. (2024). A Secure Multi-Authority Attribute Based Encryption Approach for Robust Smart Grids. *Concurr. Comput. Pract. Exp.*, 36(7). Retrieved from <https://doi.org/10.1002/cpe.7972> doi: 10.1002/CPE.7972
- Schneier, B. (1993). Description of a New Variable-Length Key, 64-bit Block Cipher (Blowfish). In R. J. Anderson (Ed.), *Fast software encryption, cambridge security*

- workshop, cambridge, uk, december 9-11, 1993, proceedings* (Vol. 809, pp. 191–204). Springer. Retrieved from https://doi.org/10.1007/3-540-58108-1_24
doi: 10.1007/3-540-58108-1_24
- Schneier, B., Kelsey, J., Whiting, D., Wagner, D. A., & Hall, C. (1998). On the Twofish Key Schedule. In S. E. Tavares & H. Meijer (Eds.), *Selected areas in cryptography '98, sac'98, kingston, ontario, canada, august 17-18, 1998, proceedings* (Vol. 1556, pp. 27–42). Springer. Retrieved from https://doi.org/10.1007/3-540-48892-8_3
doi: 10.1007/3-540-48892-8_3
- Sehrawat, D., & Gill, N. S. (2018). Lightweight Block Ciphers for IoT-based Applications: A Review. *International Journal of Applied Engineering Research*, 13(5). Retrieved from https://www.ripublication.com/ijaer18/ijaerv13n5_26.pdf
- Sehrawat, V. S., Yeo, F. Y., & Desmedt, Y. (2021). Extremal Set Theory and LWE Based Access Structure Hiding Verifiable Secret Sharing with Malicious-Majority and Free Verification. *Theor. Comput. Sci.*, 886, 106–138. Retrieved from <https://doi.org/10.1016/j.tcs.2021.07.022> doi: 10.1016/J.TCS.2021.07.022
- Shamir, A. (1979). How to Share a Secret. *Commun. ACM*, 22(11), 612–613. Retrieved from <https://doi.org/10.1145/359168.359176> doi: 10.1145/359168.359176
- Shin, Y., Koo, D., & Hur, J. (2017). A Survey of Secure Data Deduplication Schemes for Cloud Storage Systems. *ACM Comput. Surv.*, 49(4), 74:1–74:38. Retrieved from <https://doi.org/10.1145/3017428> doi: 10.1145/3017428
- Shivhare, A., Maurya, M. K., Sarif, J., & Kumar, M. (2022). A Secret Sharing-based Scheme for Secure and Energy Efficient Data Transfer in Sensor-based IoT. *J. Supercomput.*, 78(15), 17132–17149. Retrieved from <https://doi.org/10.1007/s11227-022-04533-0> doi: 10.1007/S11227-022-04533-0
- Sikder, A. K., Petracca, G., Aksu, H., Jaeger, T., & Uluagac, A. S. (2018). A Survey on Sensor-based Threats to Internet-of-Things (iot) Devices and Applications. *CoRR*, *abs/1802.02041*. Retrieved from <http://arxiv.org/abs/1802.02041>
- Smart, N., Baron, J. W., Saravanan, S., Brandt, J., & Mashatan, A. (2024). Multiparty Computation: To Secure Privacy, Do the Math: A Discussion with Nigel Smart, Joshua W. Baron, Sanjay Saravanan, Jordan Brandt, and Atefeh Mashatan. *ACM Queue*, 21(6), 78–100. Retrieved from <https://doi.org/10.1145/3639448> doi: 10.1145/3639448

- Stinson, D. R. (2004). *Combinatorial Designs - Constructions and Analysis*. Springer.
- Stinson, D. R., & Wei, R. (2018). Combinatorial Repairability for Threshold Schemes. *Des. Codes Cryptogr.*, 86(1), 195–210. Retrieved from <https://doi.org/10.1007/s10623-017-0336-6> doi: 10.1007/S10623-017-0336-6
- Tang, Z. (2021). Secret Sharing-based IoT Text Data Outsourcing: A Secure and Efficient Scheme. *IEEE Access*, 9, 76908–76920. Retrieved from <https://doi.org/10.1109/ACCESS.2021.3075282> doi: 10.1109/ACCESS.2021.3075282
- Tassa, T. (2007). Hierarchical Threshold Secret Sharing. *J. Cryptol.*, 20(2), 237–264. Retrieved from <https://doi.org/10.1007/s00145-006-0334-8> doi: 10.1007/S00145-006-0334-8
- Čučík, P., Ploszek, R., & Zajac, P. (2022). Practical Use of Secret Sharing for Enhancing Privacy in Clouds. *Electronics*, 11(17). Retrieved from <https://www.mdpi.com/2079-9292/11/17/2758> doi: 10.3390/electronics11172758
- Verheul, E. R., & van Tilborg, H. C. A. (1997). Constructions and Properties of k out of n Visual Secret Sharing Schemes. *Des. Codes Cryptogr.*, 11(2), 179–196. Retrieved from <https://doi.org/10.1023/A:1008280705142> doi: 10.1023/A:1008280705142
- Wagner, G. (1997). Multi-level Security in Multiagent Systems. In P. Kandzia & M. Klusch (Eds.), *Cooperative information agents, first international workshop, cia' 97, kiel, germany, february 26-28, 1997, proceedings* (Vol. 1202, pp. 272–285). Springer. Retrieved from https://doi.org/10.1007/3-540-62591-7_40 doi: 10.1007/3-540-62591-7_40
- Wang, N., Fu, J., Zhang, S., Zhang, Z., Qiao, J., Liu, J., & Bhargava, B. K. (2023). Secure and Distributed IoT Data Storage in Clouds Based on Secret Sharing and Collaborative Blockchain. *IEEE/ACM Trans. Netw.*, 31(4), 1550–1565. Retrieved from <https://doi.org/10.1109/TNET.2022.3218933> doi: 10.1109/TNET.2022.3218933
- Xu, J., Huang, R., Huang, W., & Yang, G. (2009). Secure Document Service for Cloud Computing. In M. G. Jaatun, G. Zhao, & C. Rong (Eds.), *Cloud computing, first international conference, cloudcom 2009, beijing, china, december 1-4, 2009. proceedings* (Vol. 5931, pp. 541–546). Springer. Retrieved from https://doi.org/10.1007/978-3-642-10665-1_49 doi: 10.1007/978-3-642-10665-1_49
- Xu, K., Yue, H., Guo, L., Guo, Y., & Fang, Y. (2015). Privacy-Preserving Machine Learning

- Algorithms for Big Data Systems. In *35th IEEE international conference on distributed computing systems, ICDCS 2015, columbus, oh, usa, june 29 - july 2, 2015* (pp. 318–327). IEEE Computer Society. Retrieved from <https://doi.org/10.1109/ICDCS.2015.40> doi: 10.1109/ICDCS.2015.40
- Yao, Y. D., & Cheng, S. (1986). Generalization of Hadamard Matrices and a Class of Two-Dimensional Error-Correcting Codes. In *IEEE international conference on communications: Integrating the world through communications, ICC 1986, toronto, canada, june 22-25, 1986, proceedings* (pp. 997–1001). IEEE.
- Zhang, A., & Lin, X. (2018). Towards Secure and Privacy-Preserving Data Sharing in e-Health Systems via Consortium Blockchain. *J. Medical Syst.*, 42(8), 140:1–140:18. Retrieved from <https://doi.org/10.1007/s10916-018-0995-5> doi: 10.1007/S10916-018-0995-5
- Zhong, H., Sang, Y., Zhang, Y., & Xi, Z. (2019). Secure Multi-Party Computation on Blockchain: An Overview. In H. Shen & Y. Sang (Eds.), *Parallel architectures, algorithms and programming - 10th international symposium, PAAP 2019, guangzhou, china, december 12-14, 2019, revised selected papers* (Vol. 1163, pp. 452–460). Springer. Retrieved from https://doi.org/10.1007/978-981-15-2767-8_40 doi: 10.1007/978-981-15-2767-8_40