Survey on Security of Blind Signature and Impossibility of Blind Signature from Lossy-Trapdoor-Permutations

Report Submitted
by
Nilanjan Datta
M.Tech (Computer Science)
Roll No. CS0901
Indian Statistical Institute

AS A FULFILLMENT OF THE DISSERTATION

Under The Guidance
of
Prof. Rana Barua
and
Mr. Rishiraj Bhattacharyya

Indian Statistical Institute
203 B.T. Road, Kolkata : 700108

July 22, 2011

Indian Statistical Institute

203, B.T. Road. Kolkata : 700108

CERTIFICATE

This is to certify that the thesis entitled "**Survey on Security of Blind Signature and Impossibility of Blind Signature Schemes from Lossy-Trapdoor-Permutations** " is submitted in the partial fulfilment of the degree of M.Tech. in Computer Science at Indian Statistical Institute, kolkata. It is fully adequate, in scope and quality as a dissertation for the required degree.

The thesis is faithfully record of bonafide research work carried out by Nilanjan Datta under our supervision and guidance. It is further certified that no part of this thesis has been submitted to any other university or Institute for the award of any degree or diploma.

Prof. Rana Barua
(Supervisor)

Mr. Rishiraj Bhattacharyya
(Supervisor)

Countersigned
(External Examinar)
Date:

Acknowledgement

I take this opportunity to express my sincere gratitude to the supervisor of this dissertation work, Prof. Rana Barua. His instructions guided me towards the learning process and the analysis and also helped me in all other aspects required for thesis. He has always been really motivating and inspiring for me throughout my dissertation. He has also given me the full freedom to think and explore new ideas and implementing those ideas.

I also take the opportunity to thank Rishiraj da, for guiding me towards the discovery of problem and giving his continuous suggestions, motivation, encouragement and helping me for the preparation of this manuscript.
I would also like to thank all my teachers who have taught me throughout the semesters in M.Tech course, specially Kishan Sir and Sumit da for teaching and giving me the basic knowledge of Cryptology, which gave me the confidence to work in this field. I also thank all my family and friends for their endless support.

Place: Kolkata

Date: Nilanjan Datta

Abstract

Blind Signature is a special form of digital signature, where the signer remains oblivious about the message, he signs and at the same time, the user can not generate any signature without the help of the signer. A seminal result in Cryptography is that signature schemes can be constructed (in a black-box way fashion) using Lossy-Trapdoor Permutations (LTDPs). In this thesis, we survey on the security of blind signature schemes in various models like in Random Oracle Model, Common Reference String Model and in Standard Model. We also survey on the security of special types of Blind Signature schemes like Partial Blind Signature scheme and Universally Composable blind Signature schemes and consider techniques like Blind Signature under Abort. Then we considered Lossy Trapdoor Permutations and it's importance in Cryptology and then proved that, there can not be any black-box constructions of Blind Signature schemes from Lossy Trapdoor Permutations (LTDPs).

Contents

Chapter 1

Introduction

In this chapter, we'll first introduce digital signature and it's importance in public key Cryptography. Then we'll claim that, in some application digital signature is not enough and introduce blind signature scheme to overcome those situations. Next, we'll formally define blind signature scheme, give some examples of blind signature schemes and give real life applications of blind signature schemes. Then we state the main result of the thesis and some required preliminaries.

1.1 Digital Signature Scheme

A digital signature scheme is a mathematical scheme for demonstrating the authenticity of a digital message or document. A valid digital signature gives a recipient reason to believe that the message was created by a known sender, and that it was not altered in transit. Digital signatures are commonly used for software distribution, financial transactions, and in other cases where it is important to detect forgery or tampering. A digital signature scheme typically consists of three algorithms: a Key Generation algorithm that outputs a private key and a corresponding public key, a Signing algorithm that, given a message and a private key, produces a signature and a Verification algorithm that, given a message, public key and a signature, either accepts or rejects the message's claim to authenticity.

The most common reasons for applying a digital signature to communications are :

- Authentication : Digital signatures can be used to authenticate the

source of messages. When ownership of a digital signature secret key is bound to a specific user, a valid signature shows that the message was sent by that user.

- Integrity : If a message is digitally signed, any change in the message after signature will invalidate the signature. So, receiver can conclude that message has been tampered.
- Non-repudiation : By this property a signer who has signed some information, cannot deny that he had signed it, at a later time.

These are various facilities given by a digital signature. Now, notice one thing, that the content of the message is revealed to the signer for generating the signature. So, we can not apply simple digital signature schemes in applications like E-Voting where the vote needs to be valid as well as should not be revealed to any one, not even the signer. To overcome this problem, a new form of signature namely blind signature, is introduced.

1.2 Basic Idea of blind signature

In cryptography, blind signature, as introduced by David Chaum [23], is a form of digital signature in which the content of a message is disguised (blinded) before it is signed. The resulting blind signature can be publicly verified against the original, unblinded message in the manner of a regular digital signature. Blind signatures are typically employed in privacy-related protocols where the signer and message author are different parties. Examples include cryptographic election systems and digital cash schemes.

An often-used analogy to the cryptographic blind signature is the physical act of enclosing a message in a special write-through-capable envelope, which is then sealed and signed by a signing agent. Thus, the signer does not view the message content, but a third party can later verify the signature and know that the signature is valid within the limitations of the underlying signature scheme. Blind signatures can also be used to provide unlinkability, which prevents the signer from linking the blinded message it signs to a later un-blinded version that it may be called upon to verify. In this case, the signer's response is first "un-blinded" prior to verification in such a way that the signature remains valid for the un-blinded message. This can be useful in schemes where anonymity is required.

Blind signature schemes can be implemented using a number of common public key signing schemes, for instance RSA. To perform such a signature, the message is first "blinded", typically by combining it in some way with a random "blinding factor". The blinded message is passed to a signer, who then signs it using a standard signing algorithm. The resulting message, along with the blinding factor, can be later verified against the signer's public key. In some blind signature schemes, such as RSA, it is even possible to remove the blinding factor from the signature before it is verified. In these schemes, the final output (message/signature) of the blind signature scheme is identical to that of the normal signing protocol.

1.3 Motivation behind Blind Signature Scheme

Blind signature schemes see a great deal of use in applications where sender privacy is important. This includes various "digital cash" schemes and voting protocols. For example, the integrity of some electronic voting system may require that each ballot be certified by an election authority before it can be accepted for counting, this allows the authority to check the credentials of the voter to ensure that they are allowed to vote, and that they are not submitting more than one ballot. Simultaneously, it is important that this authority not learn the voter's selections. An unlinkable blind signature provides this guarantee, as the authority will not see the contents of any ballot it signs, and will be unable to link the blinded ballots it signs back to the un-blinded ballots it receives for count. Here we discuss Electronic cash and Electronic Voting – the main two applications of blind signatures in details.

1.3.1 Electronic Cash

As early as 1982, Chaum's [23] pioneering work aimed at creating an electronic version of money. To achieve this goal, he introduced the notions of "coins" and "randomized blind signatures" (or simply "blind signatures"). He claimed that this was the only way to ensure the required anonymity: in real life, a coin cannot be easily traced from the bank to the shop, furthermore, two spendings of a same user cannot be linked together. These are two main properties of real coins that Chaum wanted to mimic: untraceability and unlinkability. He proposed to define an electronic coin as a number with

a certificate (a signature) produced by the bank, it is withdrawn from the bank, spent by the user, and deposited by the shop (see Fig.1.1)

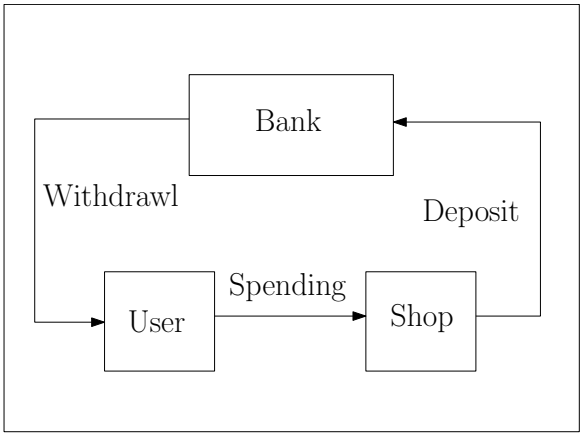

Figure 1.1: Coin Life

On-line electronic cash : In his first scheme, Chaum used blind signatures for the production of coins. The user makes the bank blindly sign a coin. Then the user is in possession of a valid coin that the bank itself cannot recognize nor link with the user. When the user spends the coin, the shop immediately returns it to the bank. If the coin has already been spent, the bank detects the fact and informs the shop so that it refuses payment. It is an "on-line" context: there is a continuous communication between the shop and the bank in order to verify the validity of coins. In order to define the scheme, Chaum introduced the first blind signature scheme, based on the RSA hypothesis. It is a by now classical transformation of the original RSA signature scheme.

Off-line electronic cash and the "cut-and-choose" methodology : In an "off-line" context one cannot prevent a user from spending a coin twice or even more, since the detection is made too late to refuse payment. This fraud is called "doublespending." We only can hope that the double-spender will be discovered later and punished. Chaum et al. [24] were able to build such schemes by introducing the identity of the user in the coin in such a way that it remains concealed, unless double-spending happens. Once, blind signatures were a critical point for anonymity, and, as before, the authors

used the blind RSA signature, together with the "cut-and-choose" technique: in their proposition, a coin is a kind of list of k blind signatures, each having an embedded copy of the identity of the user. To be sure that double-spending will reveal the real identity of the user, the bank would like to verify that the signatures actually have the requested format, which would revoke anonymity. Then the bank helps the user to get $2k$ signatures, randomly chooses k of them, and verifies the inner structure of the selected signatures. Since these signatures are no longer anonymous, the user throws them away and constructs the coin with the k other ones. The probability for a cheater to be finally in possession of a fraudulous coin is about 2^{-2k}. The main drawback of the "cut-and-choose" technique is that the coins are very large, as well as the amount of computations.

In 1993 Ferguson [28] and Brands [15] proposed new schemes without "cut-and-choose." The first one uses once again the blind RSA signature, whereas Brands' scheme uses a new blind signature derived from the Schnorr signature scheme. In both schemes Ferguson and Brands managed to hide the identity of the user in a much more efficient way than the "cut-and-choose" methodology. Again, the identity is revealed after double-spending. Those blind signatures which hide a specific structure, such as the identity, are called "restrictive blind signatures".

1.3.2 Electronic Voting System

With a rapid growth in computer networks, many people can access the network through the Internet and therefore an electronic voting can be a viable alternative for conducting an election. Electronic voting system must attempt to achieve at least the same level of security as ordinary elections. A prototype of EVS, called E-Voting, is a type of EVS, that satisfies four security requirements for a safe election : Confidentiality, Integrity, Authentication and Verifiability. Confidentiality means, the voter's ballot should be kept confidential. Integrity demands that only valid votes are counted in the final tally. Authentication ensures a voter who is allowed to vote must be an eligible voter and by verifiability, a voter can check that his vote was properly received and has been taken into account in the final tally.

Blind Signature is the most popular cryptographic technique in Electronic Voting System that is used to provide confidentiality of the voter's ballot.

The signature is used to authenticate the voter without disclosing the content of a ballot. Hence the authority whose function is to verify the eligibility of a voter will not know whom a voter votes for.

1.4 Definition (Blind Signature Scheme)

To define blind signatures formally we introduce the following notation for interactive execution between algorithms $\mathcal{X}$ and $\mathcal{Y}$. By $(a, b) \leftarrow \langle \mathcal{X}(x), \mathcal{Y}(y) \rangle$ we denote the joint execution, where x is the private input of $\mathcal{X}$, y defines the private input for $\mathcal{Y}$, the private output of $\mathcal{X}$ equals a, and the private output of $\mathcal{Y}$ is b. We write $\mathcal{Y}^{<\mathcal{X}(x),\cdot>^{\infty}}(\mathrm{y})$ if $\mathcal{Y}$ can invoke an unbounded number of executions of the interactive protocol with $\mathcal{X}$ in sequential order. Accordingly, $\mathcal{X}^{<\cdot,\mathcal{Y}(y_0)>^1,<\cdot,\mathcal{Y}(y_1)>^1}(\mathrm{x})$ if $\mathcal{Y}$ can invoke sequentially ordered executions with $\mathcal{Y}(y_0)$ and $\mathcal{Y}(y_1)$, but interact with each algorithm only once.

Definition 1. *A blind signature scheme consists of a tuple of effcient algorithms* $BS = (KG, \langle \mathcal{S}, \mathcal{U} \rangle, Vf)$ *where*

Key Generation. $KG(1^n)$ *generates a key pair* (sk, pk).

Signature Issuing. *The joint execution of algorithm* $\mathcal{S}(sk)$ *and algorithm* $\mathcal{U}(pk, m)$ *for message* $m \in \{0,1\}^n$ *generates an output* σ *of the user,* $(\perp, \sigma) \leftarrow \langle \mathcal{S}(sk), \mathcal{U}(m, pk) \rangle$

Verification. $Vf\,(pk, m, \sigma)$ *outputs a bit.*

It is assumed that the scheme is complete, i.e., for any $(\mathrm{sk},\mathrm{pk}) \rightarrow \mathrm{KG}(1^k)$, any message $\mathrm{m} \in \{0,1\}^n$ and any σ output by $\mathcal{U}$ in the joint execution of $\mathcal{S}(\mathrm{sk})$ and $\mathcal{U}(\mathrm{pk},\mathrm{m})$ we have $\mathrm{Vf}(pk, m, \sigma) = 1$.

1.5 Definition (Secure Blind Signature Scheme)

Security of blind signature schemes requires two properties, unforgeability and blindness [35, 48]. A malicious user $\mathcal{U}^*$ against unforgeability tries to generate $k+1$ valid message-signatures pairs after at most k completed interactions with the signer, where the number of interactions is adaptively determined by the user during the attack. The blindness condition says that it should be infeasible for a malicious signer $\mathcal{S}^*$ to decide upon the order in which two messages m_0 and m_1 have been signed in two executions with an honest user $\mathcal{U}$.

Definition 2. *A blind signature scheme* $BS = (KG, \langle \mathcal{S}, \mathcal{U} \rangle, Vf)$ *is called secure if the following holds:*

i. Unforgeability : *For any effcient algorithm* $\mathcal{U}^*$ *the probability that experiment* $Forge^{BS}_{\mathcal{U}^*}$ *evaluates to 1 is negligible (as a function of* n*) where*

Experiment $\boldsymbol{Forge^{BS}_{U^*}}$ *:*
$(sk, pk) \leftarrow KG(1^n)$
$((m_1, \sigma_1), ..., (m_{k+1}, \sigma_{k+1})) \leftarrow \mathcal{U}^{*\langle \mathcal{S}(sk), \cdot \rangle^{\infty}}(pk)$
Return 1 iff
$m_i \neq m_j$ *for* $1 \leq i < j \leq k+1$ *and*
$Vf\,(pk,\, m_i,\, \sigma_i) = 1$ *for all* $i = 1, 2, ..., k+1$ *and at most* k *interactions with* $\langle \mathcal{S}(sk), \cdot \rangle^{\infty}$ *were completed.*

ii. Computational resp. Statistical Blindness : *For any (effcient resp. unbounded) algorithm* $\mathcal{S}^*$ *working in modes find, issue and guess, the probability that the following experiment* $Blind^{BS}_{\mathcal{S}^*}$ *evaluates to 1 is negligibly close to* $1/2$*, where*

Experiment $\boldsymbol{Blind^{BS}_{S^*}}$ *:*
$(p, m_0, m_1, st_{find}) \leftarrow \mathcal{S}^*(find, 1^n)$
$b \leftarrow \{0, 1\}$
$st_{issue} \leftarrow \mathcal{S}^{*<\cdot, \mathcal{U}(pk, m_b)>^1, <\cdot, \mathcal{U}(pk, m_{1-b})>^1}(issue, st_{find})$
and let σ_b, σ_{1-b} *denote the (possibly undefined) local outputs of* $\mathcal{U}(pk, m_b)$ *resp.* $\mathcal{U}(pk, m_{1-b})$*.*
set $(\sigma_0,\, \sigma_1) = (\perp, \perp)$ *if* $\sigma_0 = \perp$ *or* $\sigma_1 = \perp$
$b^* \leftarrow \mathcal{S}^*(guess, \sigma_0, \sigma_1, st_{issue})$
Return 1 iff $b = b^*$*.*

We remark that, even if occassionally not mentioned, all algorithms receive the security parameter 1^n as additional input.

1.6 Examples (Blind Signature Scheme)

Many blind signature schemes have been proposed in the literature, e.g., [23, 35, 48, 2, 28] with varying characteristics of security and efficiency. Here, examples of two basic blind-signature schemes are given.

1.6.1 The Blind FDH-RSA Signature :

Here we present one of the modified version of RSA blind signature scheme, namely FDH-RSA blind signature scheme. We will first describe the scheme and then claim it's security in random oracle model.

- **Key Generation.** Let p and q be two large primes each of k bits, where k is the security parameter. let, $N = pq$. Hence $\phi(N) = (p-1).(q-1)$. A random no. $e \in {\mathbb{Z}_n}^*$ is chosen s.t. gcd(e, $\phi(N)$) = 1. Choose d s.t. $e.d \equiv 1 \bmod \phi(N)$.
  Public key = (N, e)
  Secret key = (p, q, d)

- **Signature Issuing.** Let the user wanst to get signature on the message m. Then the user choose a random no. $r \in {\mathbb{Z}_n}^*$ and computes $m' = H(m).r^e \bmod N$ and send it to signer. Here, H is a Hash function known to both the party during initial set-up.

  The Signer generate signature $\sigma^{'} = (m^{'})^d$ on the message $m^{'}$. and send it to the user. The user then computes $\sigma = \sigma^{'}.r^{-1}$.
  Now, (m, σ) is a valid message-signature pair.

- **Verification.** (m, σ) is a valid message-signature pair if $\sigma^e = H(m)$.

One can get a proof of unforgeability for this scheme, in the random oracle model, under the assumption that the RSA known-target inversion problem is hard. From [23] one can easily verify that, the blindness condition is satisfied.

1.6.2 The Blind Schnorr Signature :

Now, we give the details of another blind signature scheme in random oracle model namely blind Schnorr scheme. Like the previous one, we first describe the scheme and then claim it's security.

- **Key Generation.** The generation algorithm produces two large prime numbers p and q such that q divides $p-1$ as well as an element $g \in \mathbb{Z}_p^*$ of order q. A random no. $x \in \mathbb{Z}_q^*$ is chosen. Then y is computed s.t. $y = g^{-x} \bmod p$
  Public Key = (y).
  Secret Key = (x).

- **Signature Issuing.** In order to get the signature of a secret message m, the user asks the signer to initiate a communication. He chooses a random $K \in \mathbb{Z}_q^*$, computes and sends a commitment $r = g^K \bmod p$. The user then blinds this value with two random elements $\alpha, \beta \in Z_q^*$, into $r^{'} = r.g^{-\alpha}.y^{-\beta} \bmod p$, computes the value $e^{'} = H(m, r^{'}) \bmod q$ and sends the "challenge" $e = e^{'} + \beta \bmod q$ to the signer who returns the value s such that $g^s.y^e = r \bmod p$. Finally, the user computes $s^{'} = s - \alpha \bmod q$.

- **Verification.** The pair $(e^{'}, s^{'})$ is a valid Schnorr signature of m since it satisfies $e^{'} = H(m, g^{s^{'}}.y^{e^{'}} \bmod p)$.

The proof of the unforgeability for the scheme, in random oracle model, is given under the assumption that discrete log problem is hard. It is easy to verify that the blindness property also hold.

1.7 Main Result

There is no black-box construction of blind signature schemes from Lossy Trapdoor Permutations (LTDPs).

At a high level, our approach is similar to the one used by Dominique Schroder in the context of ruling out constructions of blind signatures from one-way functions [37], which in turn require the basic frameworks used by Barak and Mahmoody-Ghidary in the context of of black-box constructions of (standard) signature schemes from one-way functions [7]. The result imposes no restrictions on the blind signature scheme, and applies even to schemes with imperfect completeness. Moreover, the impossibility result even rules out constructions of blind signature schemes for 1-bit messages that achieve security only against honest-but-curious parties.

1.8 Thesis Organization

The remainder of this thesis is organized as follows. In chapter 2, we present a survey on various Blind Signature schemes in Random Oracle Model, Standard Model, Common Reference String Model and discuss on the security aspect. Moreover, we have also surveyed techniques like Blind Signature

under Aborts, Partial Blind Signature schemes etc. We prove the impossibility of Blind Signatures from Lossy-Trapdoor-Permutations in chapter 3. In chapter 4, we conclude with a brief discussion of the significance of the thesis and an open question in the field of Blind Signature and security issues.

Chapter 2

Survey on Existing Blind-Signature Schemes and their Security

The various blind signature schemes differ in round complexity, underlying computational assumptions, and the model in which the security proof is given. For example, many schemes [2, 48] rely on the Random Oracle heuristic, where a hash function is considered as a truly random function. It is well-known, however, that a security proof in the random oracle model does not necessarily imply security in the standard model when a random oracle is instantiated by an efficient hash function. Therefore, alternative solutions are necessary. Several blind signature schemes that achieve security in the standard model have also been proposed. These instantiations differ in the underlying number-theoretic assumptions and their round complexities. Constructions based on general assumptions are also known [35], but the minimal assumptions in terms of round complexity and computational assumptions without assuming setup assumptions are unknown.

In this section, we give a brief introduction to various Blind Signature Schemes proposed in Random Oracle, Standard and Common-Reference String Model and discuss about their security aspects. We also mentioned the important security results that were proved on various Blind Signature Schemes, on various models along with a proof idea for each of them. Then we give a notion of special Blind-Signature schemes like Partial Blind-Signature Scheme, Blind Signatures with Aborts, Universally Composable Blind Sig-

nature schemes and discuss about various security results, proved on these schemes.

2.1 Security of Blind Signature Schemes in the Random Oracle Model

In Random Oracle Model, all the algorithms of the blind-signature scheme – $KG, \mathcal{S}, \mathcal{U}$, Vf gets a black-box access to an oracle $\mathcal{O} : \{0,1\}^n \rightarrow \{0,1\}^n$, and can query the oracle on any input x. Initial constructions of blind signature schemes were in the random oracle model [9], and, in fact, until 2004, all efficient constructions relied on random oracles.

Pointcheval and Stern [49] were the first to give a secure blind signature schemes. They first proposed various definitions of unforgeability –

- (The $(l,\ l+1)$-Forgery). For any integer l, an $(l,\ l+1)$-Forgery comes from an attacker that provides $l+1$ signatures after l interactions with the signer Σ.
- (The "One-More" Forgery). A "One-More" Forgery is an $(l,\ l+1)$-Forgery for some integer l, polynomially bounded in the security parameter k.
- (The Strong "One-More" Forgery). A Strong "One-More" Forgery is an $(l, l+1)$-Forgery for some integer l, polylogarithmically bounded in the security parameter k i.e. $l \leq (logk)^{\alpha}$ for some constant α.

They also focus on two kinds of attacks –

- **The sequential attack :** the attacker interacts sequentially with the signer. This attack can be performed by a user who withdraws coins, one after the other.
- **The parallel attack :** the attacker interacts l times in parallel with the signer. This attack is stronger. Indeed, the attacker can initiate new interactions with the signer before previous ones have ended. This attack can be performed by a group of users who withdraw many coins at the same time.

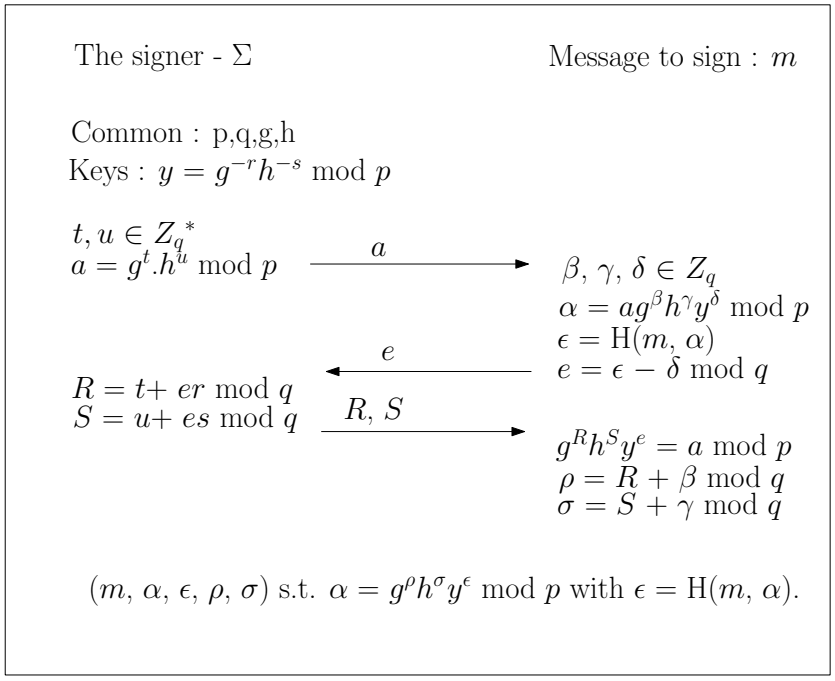

Figure 2.1: Okamoto-Schonorr Blind-signature scheme

Then, they showed the security of a certain type of efficient blind signature in the random oracle model [48]. Namely, they showed security of Okamoto-Schnorr. They use the concept of the "Witness Indistinguishable" proofs. In such a proof system:

- Many secret keys are associated to a same public key.
- The views of two proofs using two distinct secret keys (witnesses) associated to a same public key are indistinguishable, even from the point of view of the verifier.
- The knowledge of two distinct secret keys associated to a same public one provides the solution of a difficult problem.

Then, they gave witness indistinguishable adaptation of the Schnorr identification of Okamoto blind signature schemes and proved their security as long as the number of issued signatures are bounded logarithmically in the security parameter (in a restricted variant of the parallel setting). They proved

the security, by reducing the Strong "One-more" Forgeability to solve Discrete logarithm problem.

They also gave witness indistinguishable adaptation of Guillou-Quisquater identification of Okamoto blind signature schemes and proved their security as long as the number of issued signatures are bounded logarithmically in the security parameter (in a restricted variant of the parallel setting) similar as the previous scheme.

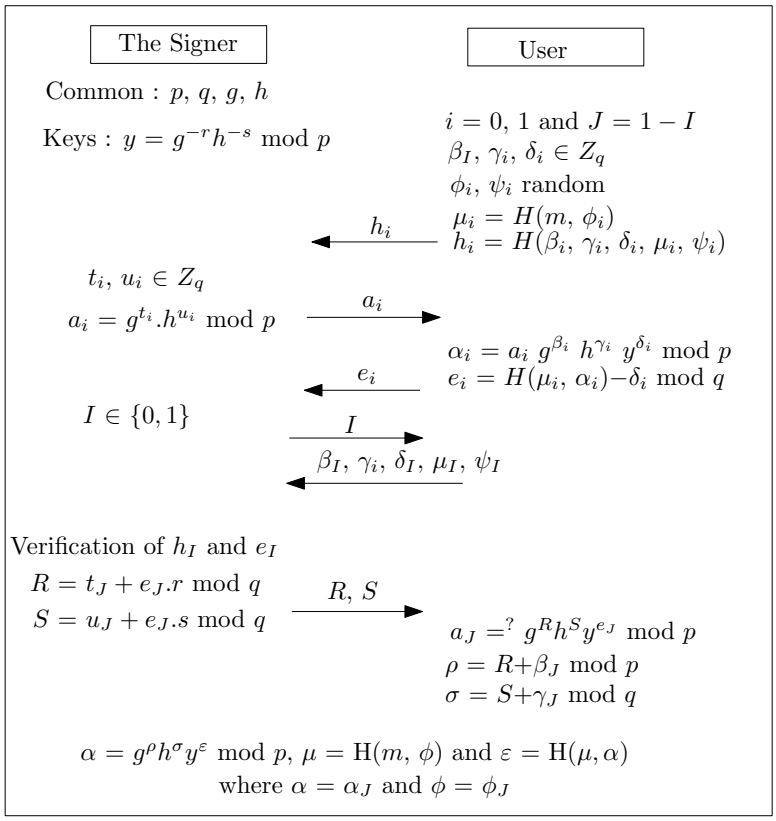

Figure 2.2: Modified Scheme by Pointcheval

Later, in [46], Pointcheval developed a generic approach that converts logarithmically secure schemes into polynomially secure ones at the cost of two more data transmissions between the signer and the receiver. With a kind of

"cut-and-choose" method, he imposed the user to play honestly. A dishonest user will be detected before it is too late.

He presented a generic transformation which makes the scheme secure after polynomially many synchronized interactions against poly-logarithmically many attackers and remains practical and efficient. In the paper, he proposed a new blind signature scheme that requires five data exchanges and proved by reduction that forgery of the new scheme under a syncronized parallel attack imply a forgery under parallel attack in the Okamoto-Schnorr Blind signature scheme using witness indistinguishability.

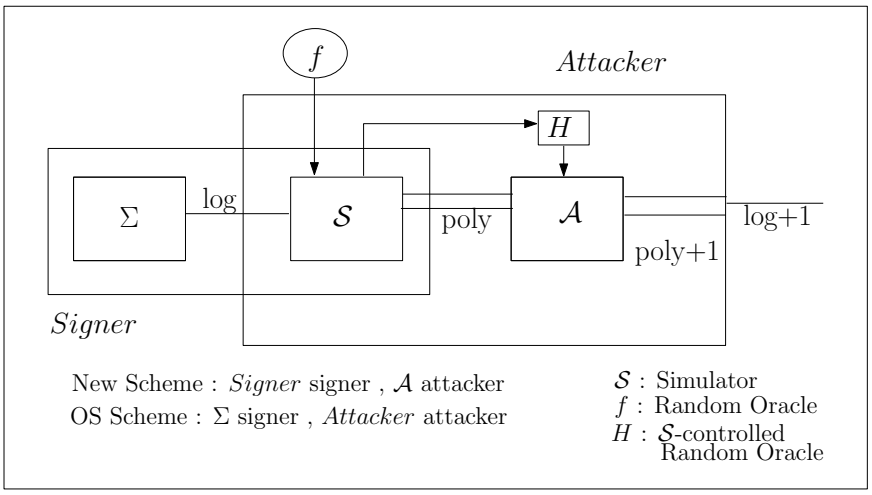

Figure 2.3: Strengthen Security for Blind-signature

He first proved that, if an adversary $\mathcal{A}$ perform an $(l, l+1)$-forgery against *Signer*, under a Synchronized parallel Attack; then Discrete logarithm problem can be solved after polynomial many calls to $\mathcal{A}$. Then he showed a reduction, from $\mathcal{A}$ performing an $(l,\ l+1)$-forgery against *Signer*, under a Synchronized parallel Attack to $\mathcal{S} \cup \mathcal{A}$ performing an $(\lambda,\ \lambda+1)$-forgery against Σ, under parallel Attack. Refer to Fig.2.1 for the reduction. By the above two steps, he proved the security of the scheme against parallel attack assuming Discrete Logarithm problem is Hard.

Till 2000, known practical blind signature schemes whose security against adaptive and parallel attacks can be proven in the random oracle model either needed five data exchanges between the signer and the user or are limited to issue only logarithmically many signatures in terms of a security parameter.

Abe [2] presented an efficient blind signature scheme that allows a polynomial number of signatures to be securely is- sued while only three data exchanges are needed. The proposed scheme is based on the partially blind signature scheme, a witness indistinguishable variant of the Schnorr signature scheme where the signer uses two public keys y(= g^x) and z(=g^w), which we call the real public key and the tag public key, respectively, in such a way that the signature can be issued only with real secret key x but no one can distinguish which secret key, i.e., x or w, was used.

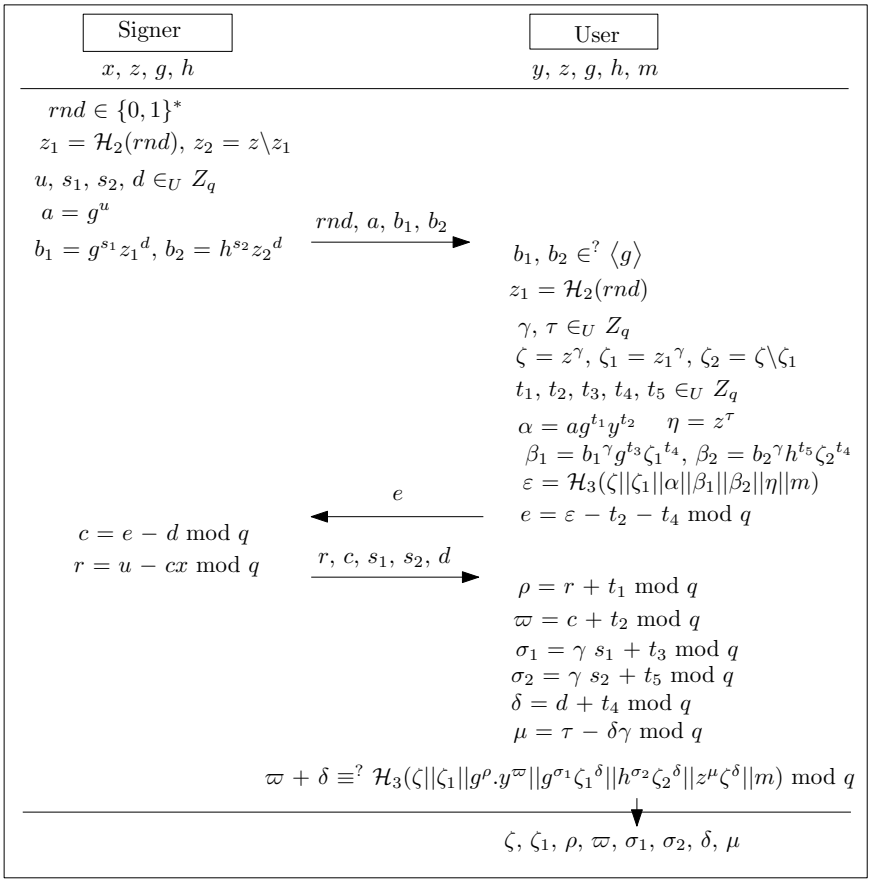

Figure 2.4: Abe's secure Blind-signature scheme

Their scheme then allows the signer to sign with several different tag public keys to achieve partial blindness. It was proven that the same tag key could be used only for logarithmically many signatures but the signer could use polynomially many tag keys. Accordingly, if the signer generates a

one-time tag key each time he signs, it achieves polynomial security, though the blindness is lost.

The scheme given by Abe provides polynomial security, i.e., one-more unforgeable even if polynomially many signatures are issued in an adaptive and concurrent manner. The security is proven in the random oracle model. The scheme remains practical as it requires only three to four times more computation than the original Schnorr signatures. Another advantage of the scheme is its potential support of protocols that need additional functionality. One can easily extend the scheme to be partially blind schemes. Furthermore, it is shown that a variant of their scheme gives a provably secure solution for double-spender-traceable electronic cash systems. Note that such e-cash schemes in the literature, rely on a variant of blind signatures called restrictive blind signatures, whose security has been proved only under non-standard and strong assumptions and only against certain restricted attacks while Abe's solution withstands the most general attacks.Thus, its security is proven in the random oracle model. Bellare, et al. [9] and Boldyreva [10] present 2-round blind signature schemes, note that 2-round protocols (which consist of a single message from the user and a response by the signer) are automatically secure in a concurrent setting.

2.2 Security of Blind Signature Schemes in the standard model

In cryptography the standard model is the model of computation in which the adversary is only limited by the amount of time and computational power available. Cryptographic schemes are usually based on complexity assumptions, which state that some problem, e.g. factorization, cannot be solved in polynomial time. Schemes which can be proven secure using only complexity assumptions are said to be secure in the standard model. Security proofs are notoriously difficult to achieve in the standard model, so in many proofs, cryptographic primitives are replaced by idealized versions.

Relatively early, it was suggested [25] that blind signatures might be constructed using protocols for generic secure 2-party computation. Juels, Luby, and Ostrovsky [35] point out that the naive way of implementing this approach does not work, but show how to adapt and extend this idea so as to

achieve a secure solution. Their approach was based on two protocols :

- Firstly, they assumed that one-way trapdoor permutation exist and there exist a polynomial time blind digital signature scheme namely Naor-Yung [43] signature scheme, which is secure against adaptive interleaved chosen message attack.

- Secondly, the two party completeness theorem which says that for any two parties A and B, where A is given a secret input x and B is given a secret input y, and any polynomial-time computable function $g(.,\ .)$ there exist a protocol for computing $g(x,\ y)$ s.t. nothing except the output of the function is revealed to the parties, moreover, the schemes could be easily extended to require that only one player learns $g(x,\ y)$, other learns nothing.

The idea that Juels et.al have was, engage the user and signer in a 2 party protocol with the Naor-Yung's signature scheme [43], at the end of which the user learns the signature of the document and signer learns nothing, thus constructing a blind signature. But this approach suffers from several problems. Juels et.al showed, how to overcome those; and finally gave the modified blind signature based on the above protocols. This was the first complexity theoratic proof of security for blind digital signatures. All previous proofs were in random oracle model only and were not fully polynomial. They showed how to achieve their protocol based on any one way trapdoor permutations. All previous schemes were based on number theoratic assumptions only. But the disadvantage of the scheme is that, although they claim security in the concurrent setting, no details of the proof in this case are provided, as best as one can tell, their solution is secure in the sequential setting only. Indeed, security of their protocol in the concurrent setting seems to require a concurrently-secure protocol for 2-party computation. Till 2004, this in fact was the only known blind signature scheme that is secure in the standard model [36] , based on general results about multi-party computation, and thus it was extremely inefficient.

In [16] Camenisch et.al present the first provably secure blind signature scheme which is also efficient. Their construction was of two steps :

- In the first step, which is a significant result on its own, they devised and proved the security of a new variant for the Cramer-Shoup-Fischlin

signature scheme, named mCSF. They showed that for generating signatures, instead of using randomly chosen prime exponents one can securely use randomly chosen odd integer exponents which significantly simplifies the signature generating process.

- In the second step based on mCFS, they obtain a blind signing function as a secure and efficient two-party computation that cleverly exploits its algebraic properties and those of the Paillier encryption scheme. This protocol is proven unforgeable only for the case of sequential attacks reling on the Strong RSA assumption and the hardness of decisional composite residuosity, they stress that it does not rely on the existence of random oracles.

Lindell [41] has shown the impossibility of concurrently-secure blind signatures if simulation-based definitions of security are used. In an effort to overcome the limitations of the above protocols, as well as Lindell's impossibility result, much recent work has focused on proving security for blind signature schemes in the concurrent setting by assuming a common reference string [38]. However, although Lindell's impossibility result was used as justification for relying on a common reference string in these works, Lindell's results do not apply if game-based security definitions (rather than simulation-based security definitions) are used.

Hazay et al. [33] present a concurrently-secure blind signature scheme and, as part of this, they also introduce a notion called a-posteriori blindness. This notion considers blindness of multiple executions between the signer and the user (as opposed to two sessions as in the basic case), and addresses the question how to deal with executions in which the user cannot derive a signature. As sketched in [33], the basic idea lies in an experiment where the adversary first outputs a public key pk together with a message distribution $\mathcal{M}$. The malicious signer then concurrently interacts with l honest user instances, where each user instance gets as input the public key pk and a message sampled according to $\mathcal{M}$. Afterwards, when the signer has finished all l interactions, it receives $l^{'}$ message-signature pairs in a randomly permuted order, where $1 \leq l^{'} \leq l$ denotes the number of non-aborted executions. The adversary wins the game if it associates one non-aborted execution to a messages-signature pair. As mentioned, the detailed discussion about a-posteriori blindness in the concurrent setting is given in [33]. The protocol

relies on standard cryptographic assumptions (e.g., trapdoor permutations and the decisional Diffie-Hellman assumption), and they prove security with respect to game-based definitions that are stronger than others that have appeared in the literature, bypassing the impossibility result of Lindell [41].

2.3 Security of Blind-Signature Schemes in Common Reference String Model

In cryptography, the common reference string (CRS) model captures the assumption that a trusted setup in which all involved parties get access to the same string crs taken from some distribution $\mathcal{D}$ exists. Schemes proven secure in the CRS model are secure given that the setup was performed correctly. The common reference string model is a generalization of the common random string model, in which $\mathcal{D}$ is the uniform distribution of bit strings.

Definition 3. *A blind signature scheme, in Common Reference String (CRS) model, consists of a tuple of effcient algorithms* $BS = (C,\ KG,\ \langle \mathcal{S},\ \mathcal{U} \rangle,\ Vf)$ *where*

CRS Generation. $C(1^n)$ *generates a common reference string* crs.

Key Generation. $KG(crs)$ *generates a key pair* $(sk,\ pk)$.

Signature Issuing. *The joint execution of algorithm* $\mathcal{S}(crs,\ sk)$ *and algorithm* $\mathcal{U}(crs,\ pk,\ m)$ *for message* $m \in \{0,1\}^n$ *generates an output* σ *of the user,*
$(\perp, \sigma) \leftarrow \langle \mathcal{S}(crs,\ sk), \mathcal{U}(crs,\ m,\ pk) \rangle$

Verification. $Vf\ (crs,\ pk,\ m,\ \sigma)$ *outputs a bit.*

It is assumed that the scheme is complete, i.e. for any $(sk,\ pk) \leftarrow \mathrm{KG}(1^k)$, any message $\mathrm{m} \in \{0,1\}^n$ and any σ output by $\mathcal{U}$ in the joint execution of $\mathcal{S}(crs,\ \mathrm{sk})$ and $\mathcal{U}(crs,\ pk,\ \mathrm{m})$ we have $\mathrm{Vf}\ (crs,\ pk,\ \mathrm{m},\ \sigma) = 1$. Schemes based on factoring related assumptions have been given in in the common reference string (CRS) model [16], schemes based on discrete logarithm related assumptions have been given in the CRS model [29], schemes based on a combination of discrete logarithm and factoring based assumption have been given in the CRS model [38], Finally in [27, 35] schemes in the CRS

are given under general assumptions. Due to the round optimal nature of a two-move signature request phase, and the desire to avoid the use of the random oracle, much recent work has focused on developing round optimal blind signatures in the CRS model.

In [27] Fischlin presented a scheme in the CRS which has a two move signature request protocol. The scheme is a generic construction from basic primitives, namely schemes for commitment, encryption and signatures as well as generic non-interactive zero knowledge (NIZK) proofs for NP-languages. The signature request protocol consists of the user sending a commitment to the message to the signer, who responds with a signature on the commitment. The user then uses this signature on the commitment to construct the blind signature, by first encrypting the commitment and the signature, and then adding a NIZK proof that the encrypted signature is a valid signature on the encrypted commitment, and that the encrypted commitment is a commitment to the specific message.

Using the notion of automorphic signatures Fuchsbauer [29], presents a variant of the construction of Fischlin, using specifc efficient components. In particular he makes use of the efficient NIZK proofs of Groth and Sahai [31] which hold for only specific NP-statements in pairing groups. In Fuchsbauer's scheme the blind signature is constructed by providing a Groth-Sahai proof of knowledge of a signature on a message (as opposed to a signature on a commitment as in Fischlin's generic construction). This makes the underlying NIZK proofs simpler, but makes use of a different signature request phase. The resulting blind signature consists of around 30 group elements, and is the most efficient round optimal blind signature scheme in the CRS known to date.

Fuchsbauer's scheme is based on a new intractibility assumption called the ADH-SDH problem, which he shows holds in the Generic Group Model (GGM). This is a falsifiable assumption, in the sense of Naor [42], which is closely related to the q-SDH problem lying behind the Boneh-Boyen signature scheme [14]. However, the resulting blind signature is not a standard signature, e.g. it is not a true BonehBoyen signature.

Then E. Ghada and N.P. Smart, present a round optimal blind signature scheme in the CRS model which is significantly more efficient than Fuchsbauer's scheme, a signature only consists of three group elements. Indeed the resulting signature is a standard Camenisch-Lysyanskaya (CL) signature on the message m [17]. Their required hardness assumption, being interactive, is not falsifiable. However, this property is inherited from the underlying CL signature where the underlying hardness assumption is the LRSW assumption.

2.4 Universal Composability Security of Blind Signatures

Canetti introduced the Universal Composability (UC) framework as a new approach for analyzing the security of cryptographic primitives and protocols [19]. In the UC framework, it is guaranteed that a secure primitive/ protocol maintains its security even if other primitives/ protocols run concurrently. Since UC security requirement is very strong, it raises the new question of whether conventional security notions satisfy UC security.

Canetti gave a positive answer to this question on digital signatures and Public Key Encryption (PKE). He showed that a UC-secure signature scheme is equivalent to a secure (existential unforgeable against chosen-message attacks) signature scheme, and that UC-secure PKE is equivalent to secure (semantically secure against chosen-ciphertext attacks) PKE [19]. On the other hand, as a negative answer, Canetti, Kushilevitz and Lindell showed that no (non-trivial) two-party protocol can be UC-secure in the plain model where we use no setup assumptions except for authenticated communication [21].

Since a Blind Signature scheme is not just a two party protocol nor a simple primitive like signatures and PKE, it is far from trivial to show the relationship between UC security and the conventional security of blind signatures. In the paper [26], Seiji Doi, Yoshifumi Manabe and Tatsuaki Okamoto showed that the conventional security of blind signatures is truly weaker than UC security. They formulated the security of blind signatures in the UC framework (i.e., define the ideal functionality of Blind Signatures), and showed

that the class of UC-secure Blind Signatures is a proper subset of that of secure (in the sense of [35]) Blind Signatures, assuming a one-way trapdoor permutation. Then, they introduced a stronger security definition (stronger blindness, SB-security) of blind signatures than that by Juels et al. [35]. SB-security is more appropriate in many applications (e.g., electronic cash and voting) than Juels et al.s. Then they also showed that SB-security of blind signatures is also truly weaker than security in the UC framework. That is, the class of UC-secure blind signatures is a proper subset of that of SB-secure blind signatures, assuming a one-way trapdoor permutation.

2.5 Security of Partial Blind Signature Schemes

Partially blind signature schemes are an extension of blind signature schemes that allow a signer to explicitly include necessary information (expiration date, collateral conditions, or whatever) in the resulting signatures under some agreement with the receiver. As partially blind signatures can be regarded as ones lying between ordinary non-blind digital signatures and fully blind signatures, they should satisfy the security requirements assigned to ordinary digital signatures and those of blind signatures.

In tradition blind signature, the signer has no control over the attributes except for those bound by the public key. For instance, if a signer issues blind signatures that are valid until the end of the week, the signer has to change his public key every week! This will seriously impact availability and performance. A similar shortcoming can be seen in a simple electronic cash system where a bank issues a blind signature as an electronic coin. Since the bank cannot inscribe the value on the blindly issued coins, it has to use different public keys for different coin values. Hence the shops and customers must always carry a list of those public keys in their electronic wallet, which is typically a smart card whose memory is very limited. Some electronic voting schemes also face the same problem when an administrator issues blind signatures to authorize ballots. Since he can not include the vote ID, his signature may be used in an unintended way. This means that the public key of the administrator must be disposable. Accordingly, each voter must download a new public key for each vote. A partially blind signature scheme allows the signer to explicitly include com- mon information in the blind signature under some agreement with the receiver. For instance, the signer

can attach the date of issue to his blind signatures as an attribute. If the signer issues a huge number of signatures in a day, including the date of issue will not violate anonymity. Accordingly, the attributes of the signatures can be decided independently from those of the public key.

In the scenario of issuing a partially blind signature, the signer and the user are assumed to agree on a piece of common information, denoted as $info$. In some applications, info may be decided by the signer, while in other applications it may just be sent from the user to the signer. Anyway, this negotiation is done outside of the signature scheme. In 2000, Masayuki Abe and Tatsuaki Okamoto [3] formalize this notion by introducing a polynomial-time deterministic function $Ag()$ which, takes two arbitrary strings $info_s$ and $info_u$ that belong to the signer and the user, respectively, and outputs $info$. To compute Ag, the signer and the user will exchange $info_s$ and $info_u$ with each other. However, if an application allows the signer to control $info$, then Ag is defined such that it depends only on $info_s$. In such a case, the user does not need to send $info_u$.

Definition 4. *A Partially blind signature scheme is a four-tuple* $(\mathcal{G}, \mathcal{S}, \mathcal{U}, \mathcal{V})$.

- $\mathcal{G}$ *is a probabilistic polynomial-time algorithm that takes security parameter* n *and outputs a public and secret key pair* (pk, sk).
- $\mathcal{S}$ *and* $\mathcal{U}$ *are a pair of probabilistic interactive Turing machines each of which has a public input tape, a private input tape, a private random tape, a private work tape, a private output tape, a public output tape, and input and output communication tapes. The random tape and the input tapes are read-only, and the output tapes are write-only. The private work tape is read-write. The public input tape of* $\mathcal{U}$ *contains* pk *generated by* $\mathcal{G}(1^n)$*,the description of* Ag*, and* $info_u$*. The public input tape of* S *contains the description of* Ag *and* $info_s$*. The private input tape of* $\mathcal{S}$ *contains* sk*, and that for* $\mathcal{U}$ *contains message* msg*. The lengths of* $info_s$*,* $info_u$*, and* msg *are polynomial in* n*.* $\mathcal{S}$ *and* $\mathcal{U}$ *engage in the signature issuing protocol and stop in polynomial-time. When they stop,the public output tape of* $\mathcal{S}$ *contains either completed or notcompleted. If it is completed,then its private output tape contains common information* $info^{(s)}$*. Similarly,the private output tape of* $\mathcal{U}$ *contains either* $\perp$ *or* $(info, msg, sig)$.

– $\mathcal{V}$ *is a (probabilistic) polynomial-time algorithm that takes* $(pk, info, msg, sig)$ *and outputs either accept or reject.*

In [3], then Masayuki Abe and Tatsuaki Okamoto have constructed an efficient partial blind signature scheme (shown in Fig. 2.5) based on the Schnorr signature scheme.

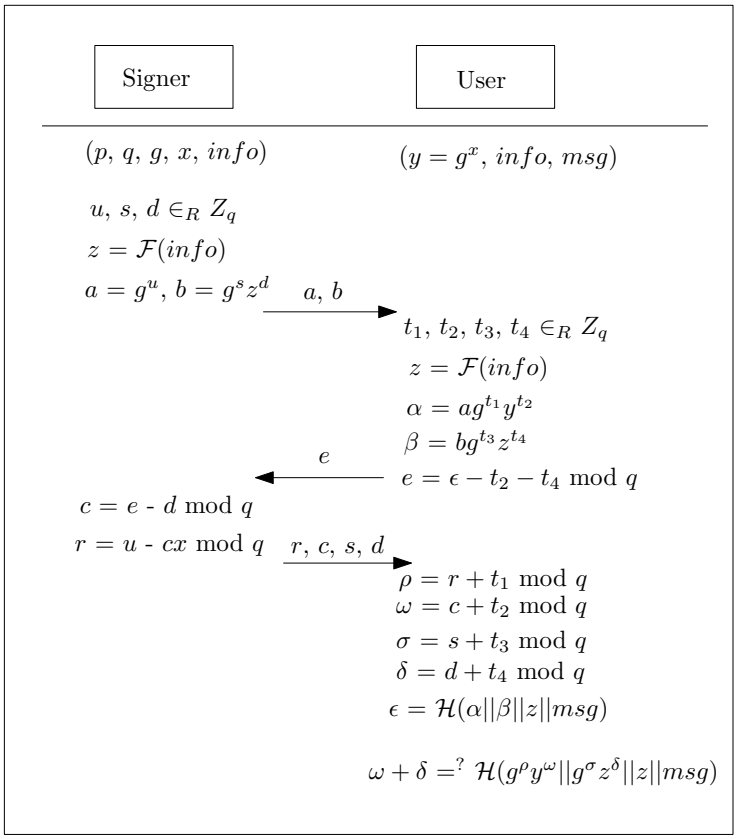

Figure 2.5: Partially Blind WI-Schnorr signature scheme

Then, they gave a proof of security in the random oracle model assuming the intractability of the discrete logarithm problem.

Since the technique developed by Pointcheval and Stern for proving the one-more-unforgeability [49] is not applicable in the scheme, they provided a new technique, applicable to variety of schemes based on the witness indistinguishable protocols, to prove the security of the scheme. As well as the

result of [48, 49] their proof guarantees that the proposed scheme is secure as long as only a logarithmic number of signatures are issued. So plugging the scheme into the generic, but yet practical scheme of [46] will yield a scheme secure up to polynomial number of signatures. For the sake of simplicity, they put off the generic description of our approach and concentrate on describing one particular scheme based on the original (i.e. not Okamoto version of) Schnorr signature scheme.

One can, however, construct a scheme in a similar way based on Guillou-Quisquater signatures [32] or variants of modified ElGamal signatures [49] at the cost of doubling the computation and communication compared to the underlying schemes. Although the primary goal was partially Blind signatures, their approach also yielded secure fully Blind signatures. One can easily transform fully blind signature schemes from partially blind ones and the reverse is also possible i.e. partially blind signature schemes can be derived from fully blind witness indistinguishable signature schemes.

2.6 Security of Blind Signature Under Abort

Blind signatures under aborts, is a technique, where the user or the signer may stop the interactive signature issue protocol prematurely. Several works on Blind signatures discuss security only in regard of completed executions and usually do not impose strong security requirements in case of aborts. One of the exceptions is the paper of Camenisch, Neven and shelat (Eurocrypt 2007).

Camenisch et al. [18] considered the limitations of the standard blindness notion. They have introduced an extension called selective-failure blindness in which the a malicious signer should not be able to force an honest user to abort the signature issue protocol because of a certain property of the user's message, which would disclose some information about the message to the signer. They define the Selective Failure blind signature as follows :

A Blind Signature scheme, BS = (KG,$\langle \mathcal{S}, \mathcal{U} \rangle$,Vf) is called Selective Failure blind, if the probability that the following experiment $SFBlind^{BS}_{S^*}(n)$ evaluates to 1 is negligibly close to 1/2, where

Experiment $\mathbf{SFBlind}^{\mathbf{BS}}_{\mathbf{S^*}}(n)$:
$(pk, m_0, m_1, st_{find}) \leftarrow \mathcal{S}^*(find, 1^n)$
$b \leftarrow \{0, 1\}$
$st_{issue} \leftarrow \mathcal{S}^{*<\cdot,\ \mathcal{U}(pk,m_b)>^1,<\cdot,\ \mathcal{U}(pk,m_{1-b})>^1}(issue, st_{find})$
and let σ_b, σ_{1-b} denote the (possibly undefined) local outputs of $\mathcal{U}(pk, m_b)$ resp. $\mathcal{U}(pk, m_{1-b})$.
define answer as : left, if only first execution has failed,
right, if only second execution has failed,
both, if both execution has failed
and (σ_0, σ_1) otherwise.
$b^* \leftarrow \mathcal{S}^*(guess, answer, st_{issue})$
Return 1 iff $b = b^*$.

Camenisch et. al. presents a construction of a simulatable oblivious transfer protocols from so-called unique selective-failure Blind signature schemes (in the random oracle model) for which the signature is uniquely determined by the message. Since the main result of the work [18] is the construction of oblivious transfer protocols, the authors note that Chaum's scheme [23] and Boldyreva's protocol [28] are examples of such selective-failure Blind schemes, but do not fully explore the relationship to (regular) blindness. Thus, selective-failure blindness does not follow from this notion. Aborts of players have also been studied under the notion of fairness in two-party and multi-party computations, especially for the exchange of signatures, e.g. [30, 5]. Fairness should guarantee that one party obtains the output of the joint computation if and only if the other party receives it. Note, however, that in case of Blind signatures the protocol only provides a one-sided output to the user (namely, the signature). In addition, solutions providing fairness usually require extra assumptions like a trusted third party in case of disputes, or they add a significant overhead to the underlying protocol.

Marc Fischlin and Dominique Schroder pick up the idea of selective-failure blindness to deal with signer aborts and expand the work of Camenisch et

al. [18] towards its relationship to blindness and further constructions of such schemes. In their paper on Security of Blind Signature Under Aborts They've done the following :

- They showed that, selective-failure blindness is indeed a strictly stronger notion than regular blindness.
- They extended the notion of selective-failure blindness to multiple executions, particularly addressing aborts of a subset of executions. They gave two possible definitions for the multi-execution case :
  1. The first definition is an ordering-based definition where the adversary has to distinguish the order of two different executions.
  2. The second definition is a prediction based one, where the malicious signer has to link an execution to a message-signature pair.

  Then proved them to be equivalent and then showed that blindness in the basic case of two executions suffices to guarantee security in the case of many sessions.
- Then, they presented a general transformation which turns every secure Blind signature scheme into a selective failure Blind scheme with an additional commitment of the message, which the user computes before the actual protocol starts and which the user then uses in the original protocol instead of the message itself. Since the commitment is non-interactive, the transformation inherits important characteristics of the underlying protocol like the number of moves and concurrent security though, the transformation destroys uniqueness (i.e. each message has only one valid signature per key pair), as required by [18] to derive oblivious transfer from such Blind signatures.
- However, they showed that the transformation was still applicable by modifing the oblivious transfer protocol of [18] slightly. Hence, they obtained an adaptive oblivious transfer from any unique Blind signature scheme such that the protocol is simulatable in presence of failures. They showed that selective-failure blindness is not necessary to obtain such oblivious transfer protocols, but uniqueness is sufficient. Their result was in the random oracle model.

- They finally studied the case of user aborts and showed that every three-move Blind signature scheme is unforgeable under user aborts. While this is clear for two-move schemes like Chaum's protocol [23] they showed that this remains true for other schemes like the ones by Pointcheval and Stern [48] but in general, this does not hold for schemes with four or more moves, assuming the existence of a secure two-move Blind signature scheme.

In summary, their transformation to achieve selective-failure blindness, together with the result about user aborts, showed that any scheme with two or three moves can be efficiently turned into one, which is secure under aborts (of either party).

2.7 Impossibility Results on 3 move Blind Signature Schemes

Marc Fischlin and Dominique Schroder investigate investigate the possibility of instantiating the random oracles in the schemes by Chaum and by Pointcheval and Stern, and of giving a security proof based on standard assumptions like RSA or discrete logarithm. Although both schemes are different in nature we can subsume them under a more general pattern of Blind signature schemes where

- Blindness holds in a statistical sense, i.e., where even an unbounded malicious signer cannot link executions of the issuing protocol to message-signature pairs,
- The interactive signature issuing has three (or less) moves, and
- One can verify from the communication between a possibly malicious signer and an honest user if the user is eventually able to derive a valid signature from the interaction.

Given a Blind signature scheme with the properties above they show that for such schemes fnding black-box reductions from successful forgers to any underlying non-interactive cryptographic problem (like RSA, discrete-log or general one-wayness or collision-resistance) is infeasible. The key idea to their result is as follows: Assuming a three-move Blind signature scheme

as above and a reduction $\mathcal{R}$ reducing unforgeability to a presumably hard problem (given only black-box access to an alleged forger). Vice versa, if the problem is indeed infeasbile, then the reduction therefore shows that the scheme is unforgeable.

Their approach is to show that the existence of a reduction $\mathcal{R}$ as above already violates the assumption about the hardness of the underlying problem. Our starting point is to design an oracle Σ with unlimited power and a "magic" adversary $\mathcal{A}$ breaking the unforgeability of the Blind signature scheme with the help of Σ. By assumption, the reduction $\mathcal{R}$ with access to $\mathcal{A}^{\Sigma}$ is then able to break the underlying cryptographic problem (see the left part of Figure 2.6). Note that, at this point, we are still in a setting with an all-powerful oracle Σ and the non-interactive problem may indeed be easy relative to this oracle, without contradicting the presumed hardness in the standard model.

Then, they apply meta-reduction techniques, to remove the oracle Σ from the scenario. Given $\mathcal{R}$ we show how to build a meta-reduction $\mathcal{M}$ (a "reduction for the reduction") to derive an efficient solver for the problem, but now without any reference to the magic adversary and Σ (right part of Figure 2.6). To this end, the meta-reduction $\mathcal{M}$ fills in for adversary A^{Σ} and simulates the adversary's actions without Σ, mainly by resetting the reduction $\mathcal{R}$ appropriately. We have then eventually derived an algorithm $\mathcal{M}^{\mathcal{R}}$ solving the underlying non-interactive problem in the standard model, meaning that the problem cannot be hard. In other words, there cannot exist such a reduction $\mathcal{R}$ to a hard problem.

They consider very general reductions running multiple instances of the adversary in a concurrent and resetting manner, covering all known reductions for Blind signatures in the literature. Yet, since the meta-reduction itself uses rewinding techniques, they somewhat need to restrict the reduction in regard of the order of starting and finishing resetted executions of different adversarial instances (called resetting with restricted cross-resets). This saves them from an exponential running time for $\mathcal{M}$. For example, any resetting reduction running only a single adversarial instance at a time obeys our restriction. At this point it seems as if they have not used the blindness property of the scheme and that the idea would paradoxically also apply to regular signature schemes (for which we know secure constructions based on

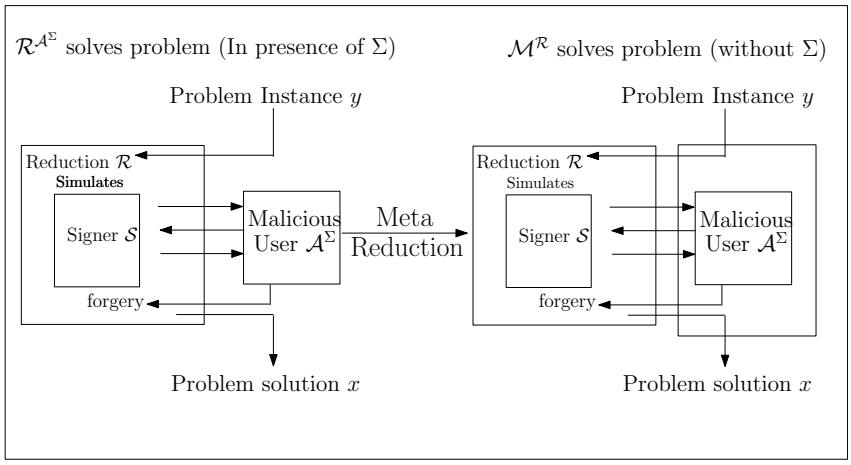

Figure 2.6: Meta Reduction Technique

any one-way function). This is not the case.

The blindness subtly guarantees that the meta reduction's simulation of the adversary is indistinguishable from the actual behavior of $\mathcal{A}^\Sigma$, such that the success probabilities of $\mathcal{R}^{\mathcal{A}^\Sigma}$ and of $\mathcal{M}^{\mathcal{R}}$ are close. For these two cases to be indistinguishable, namely $\mathcal{R}$ communicating with $\mathcal{A}^\Sigma$ or with $\mathcal{M}$, we particularly rely on the fact that blindness holds relative to the all-powerful oracle Σ used by $\mathcal{A}$, as in case of statistically-Blind signature schemes.

They have formally defined signature derivation check, which can tell whether user is able to compute a valid signature or not from public data and communication between a malicious signer and honest user and then using this idea of meta reduction, proved the following :

- There is no vanilla black-box reduction from unforgeability of the Blind signature scheme with signature derivation checks to a hard non-interactive problem.
- There is no three-move Blind signature scheme, with resetting (with restricted cross-resets) black- box reduction from unforgeability of the

Blind signature scheme BS, having the property signature derivation check, to a hard non-interactive problem.

Thus, using a technique named, Meta-reduction, they have shown impossibility results on 3 move blind signatures.

Chapter 3

Impossibility of Blind Signatures From Trapdoor Permutations (LTDPs)

3.1 Preliminaries

In this section, we first start with the definitions and security properties of trapdoor permutations. Then, we give the definition of lossy trapdoor function as given by Peikert and Waters [45]. Then, we state the importance of lossy trapdoor permutations in cryptology. Then, we give the lossy trapdoor permutation oracles and prove that, using those oracle as black-box, the impossibility result of constructing secure blind signature scheme.

3.1.1 Trapdoor Permutations (TDPs)

Definition 5. *A trapdoor permutation family is a triplet of PPTM* (Tdg, F, F^{-1}). Tdg *is probabilistic and on input* 1^n *outputs a key-pair* $(pk,\ td) \leftarrow Tdg(1^n)$. $F(pk, \cdot)$ *implements a permutation* f_{pk} *over* $\{0,1\}^n$ *and* $F^{-1}(td, \cdot)$ *implements the corresponding inverse* ${f_{pk}}^{-1}$.

3.1.2 Security Properties of Trapdoor Permutations (TDPs)

- **One-Wayness :** The most standard security property of TDP is one-wayness which says that it is hard to invert a random element without knowing the trapdoor. Formally, for any PPTM A $Pr[(pk, td) \leftarrow Tdg(1^n),\ x \leftarrow \{0,1\}^n : A(f_{pk}(x)) = x] \leq negl(n)$.

- **Partial one-Wayness :**Another standard security property of TDP is partial one-wayness which says that it is hard to invert a random element without knowing the trapdoor. Formally, for any PPTM A $Pr[(pk, td) \leftarrow Tdg(1^n),\ x \leftarrow \{0,1\}^n,\ x' \leftarrow \{0,1\}^n : A(f_{pk}(x)) = x'\ ,\ f_{pk}(x) = f_{pk}(x')] \leq negl(n)$.

- **Claw-Freeness :** Let $f_0,\ f_1$ be permutation over a common domain $\mathcal{D}$. We say that (x, y, z) is f-claw if $f_0(x) = f_1(y) = z$.

  Definition 6. *A family $\mathcal{F} = \{f_0,\ i,\ f_1,\ i : \mathcal{D}_i \rightarrow \mathcal{D}_i\ \} \in \mathcal{I}$ is called a family of **Claw-free Trapdoor Permutations** if:*

  1. *There exist an algo G such that $G(1^k)$ outputs two pairs $(f_0,\ t_0)$, $(f_1,\ t_1)$ where t_i is trapdoor information for f_i.*
  2. *There exists PPT, an algorithm that given f_i and $x \in \mathcal{D}_i$ computes $f_i(x)$*
  3. *$\forall$ (inverting algo) $\mathcal{I}$, there exists some non-negligible func nonnegl such that for all sufficiently large k,*
     $Prob[f_0(x) = f_1(y) = z : ((f_0,\ t_0),\ (f_1,\ t_1)) \leftarrow G(1^k),\ (x, y, z) \leftarrow \mathcal{I}\ (f_0,\ f_1)] < nonnegl(k)$

3.1.3 Lossy Trapdoor Permutations(LTDPs)

Lossy Trapdoor Permutations (LTDPs) were introduced by Peikert et. al. in [45]. In the paper [45], they considered a straightforward generalization to permutations.

Definition 7. *A family of (n, l) Lossy Trapdoor Permutations ($LTDPs$) is given by a Tuple $(\mathcal{S}, \mathcal{F}, \mathcal{F}')$ of PPTMs. S is a sampling algorithm which on input 1 invokes F and on input 0 invokes $\mathcal{F}'$. $\mathcal{F}$ (called "$Injective\ Mode$") describes a usual trapdoor permutation, i.e. it outputs $(f,\ f^{-1})$ where f is*

a permutation over $\{0,1\}^n$ *and* f^{-1} *is the corresponding inverse.* $\mathcal{F}'$ (*called "*$LossyMode$*") outputs a function* f' *on* $\{0,1\}^n$ *with range size at most* 2^l. *For any distinguisher* $\mathcal{D}$, *LTDP-Advantage is defined as,*

$$Adv^{ltdp}{}_{(F,F^{-1}),D} = |Pr[\mathcal{D}^f(\cdot) = 1 : (f, f^{-1}) \leftarrow \mathcal{F}] - Pr[\mathcal{D}^{f'}(\cdot) = 1 : f' \leftarrow \mathcal{F}'] \,|$$

We call $\mathcal{F}$ *"lossy" if it is the first component of some lossy LTDPs.*

3.1.4 Imporatance of Lossy Trapdoor Permutations(LTDPs) in Cryptology

Lossy Trapdoor functions, proposed by Peikert et.al. [45] have the following advantages :

- Lossy Trapdoor Permutations (LTDPs) can be realized based on the hardness of the decisional Diffie-Hellman (DDH) problem in cyclic groups, and the hardness of worst-case problems on lattices.
- Lossy Trapdoor Functions (LTDPs) imply injective (one-to-one) trapdoor functions in the traditional sense. This yields the first known trapdoor functions based on number theoretic problems that are not directly related to integer factorization.
- A black-box construction of a CCA-secure cryptosystem can be constructed based on Lossy Trapdoor Permutations (LTDPs). Peikert et.al, in their paper [45] showed it. Their approach has two main benefits: First, the construction is black-box, making it more efficient than those following the general NIZK paradigm and moreover, their's is the first known construction of a CCA-secure cryptosystem based entirely on lattice assumptions, for which there is currently no known realization in the NIZK framework.
- Moreover, Lossy Trapdoor Permutations (LTDPs) can be used to construct collision-resistant hash functions and oblivious transfer (OT) protocols, in a black-box manner. Using standard (but non-black box) transformations, this implies general secure multiparty computation for malicious adversaries.

3.1.5 Lossy Trapdoor Permutation Oracle

According to [11] , one can define a pair of oracle $(T,\ T^{'})$. Choose $2^n + 1$ permutations $\pi_0,\ldots,\pi_{2^n-1}$ and g uniformly at random from the set of all permutations over $\{0,\ 1\}^n$. Moreover choose 2^n functions $f_0, \ldots, f_{2^n-1}$ uniformly at random from the set of all functions from $\{0,\ 1\}^n$ to $\{0,\ 1\}^l$.

Oracle T works as follows:
$T_1(td) \rightarrow g(td)$ (generate public key from the trapdoor)
$T_2(pk,\ y) \rightarrow \pi_{pk}(y)$ (evaluate)
$T_3(td,\ z) \rightarrow \pi^{-1}_{g(td)}(z)$ (inversion)

On the other hand $T^{'}$ is defined as follows :
$T^{'}(pk,\ x) = \pi_{pk}(0^{n-l}||f_{pk}(x))$

Now we define the $LTDP^{T,T^{'}} = (S, (F, F^{-1}, F^{'})$ as follows :

- $S(b)$ If $b = 1$, choose a uniform random $td \leftarrow \{0,\ 1\}^n$, compute $pk = T_1(td)$ and return $(pk,\ td)$, otherwise choose a uniform random $pk \leftarrow \{0,\ 1\}^n$ and return $(pk,\ \perp)$.
- $F(pk,\ y)$ returns $T_2(pk,\ y)$.
- $F^{-1}(td,\ z)$ returns $T_3(td,\ z)$.
- $F^{'}(pk,\ y)$ returns $T^{'}(pk,\ x)$.

Lemma 1. *$LTDP^{T,T^{'}}$ implements a secure $(n,\ l)$ Lossy Trapdoor Permutation when $l = \mathcal{O}(n^{\frac{1}{c}})$ for a positive constant c.*

Proof. To show the security of $LTDP^{T,T^{'}}$, we need to argue that for any efficient distinguisher D, $|Pr[D^F = 1] - Pr[D^{F^{'}} = 1]|$ is negligible. Consider a random function $f : \{0,\ 1\}^n \rightarrow \{0,\ 1\}^l$ and a random permutation,
$\pi : \{0,\ 1\}^n \rightarrow \{0,\ 1\}^n$. It is easy to check that $\pi(0^{n-l}||f())$ has the same distribution of a random permutation until a collision in f. f being a random function, the collision probability is $q^2 \backslash 2^l$, which is negligible for $q = \mathcal{O}(n^{c_1})$ for some constant $c_1 > 0$.

Now using the fact that a function (permutation) chosen uniformly at random from the set of exponentially many functions (permutations) is indistinguishable form a random function (permutation), the lemma follows. □

Hence, for F we can use the general trapdoor permutation oracle $\pi(\cdot)$ and for F', $\pi(0^{n-l}||f(\cdot))$ can be used as an Lossy Oracle, where f is a many-to-one function, given by, $f : \{0,1\}^n \to \{0,1\}^l$.

3.2 Overview of the Proof Technique.

From a technical point of view, this proof technique is similar to that of Katz and Schroder [37]. We are interested to show that there can't be any Blind signature scheme in standard model from Lossy Trapdoor Permutation, we've chosen appropriate oracles that will implements lossy trapdoor permutations. Now, we will prove, that a blind signature scheme, using any of the two oracles, will fail to achieve the unforgeability property with non negligible probability and the views of the two prove are indistinguishible, proving the property of lossy trapdoor oracle, holds.

Katz and Scroder [37], gave the idea, how to prove that there is no black-box construction of blind signature from trapdoor permutations. Now, we have to prove the result for lossy permutation oracle and claim that, both the views are indistinguishable because otherwise, it wouldn't hold the lossy trapdoor permutation property.

Here we are giving the intuition of the proof technique that there is no blackbox construction of blind signature from trapdoor permutations, which is similar to that of [37]. An interactive signature issue protocol between a signer and a user, is considered. The input of the signer is a private key sk and the users input is a public key pk and a message m. At the end of this protocol the user outputs a signature σ on the message m. Both of the algorithms are given black-box access to the Lossy permutation oracle, given above. It is assumed that both players follow the protocol. The players are given the power to be computationally unbounded, but require that they can query the Lossy Permutation oracle, a polynomial number of times.

Now, in the setting of Blind signatures, security demands 2 properties : Unforgeability and Blindness. If schemes that only achieve property Unforgeability are considered, then it is well-known that these can be built in a black-box way from Lossy Trapdoor Permutations (LTDPs) as this is just a standard signature scheme. On the other hand, schemes that are only Blind

but trivially forgeable can be constructed without any assumption letting the verification algorithm always output 1. Our job is to show that, if one wish to satisfy both conditions above then Lossy Trapdoor Permutations (LTDPs) are not sufficient.

To illustrate the main idea why this is true, consider the setting where both the user and signer are given access to a Lossy Trapdoor Permutation Oracle. Now consider two protocol executions in which the user first obtains a signature on the message m_0 and then obtains a signature on the message m_1. We observe the following :

- Correctness intuitively requires that in each interaction the user learns sufficiently many of the queries, in order to be able to derive a valid signature.
- Unforgeability requires that the user must not learn "too many" of the queries in each interaction, that it can derive another signature on some other message. In particular, the user should not learn enough queries in the first interaction during generating signature on m_0, so that he can derive a valid signature on m_1.
- Now, Blindness implies that, from the point of view of the signer, the queries the user learns in the first interaction should be distributed identically to the queries the user learns in the second interaction.

I'll show that all these requirements are in conflict. More formally, the main idea rely on results of [8], [34] showing that for any two-party protocol there is an algorithm $Find_\delta$ that takes as input a transcript of an execution of the protocol and outputs, with high probability, a set that contains every oracle query that was asked by both parties ("intersection queries"). Viewing that the signer can run this algorithm, the blindness requirement thus implies that the set obtained by running $Find_\delta$ on the signature-issue protocol for m_0 must contain a set of intersection queries that are sufficient to derive a signature on the message m_1. Otherwise, the signer knows that the first execution could not possibly have been for m_1. Using this property, I've construct the forger, in a similar manner to the forger given in [37]. The forger executes the following steps during the attack:

1. **Set-up Algorithm and Generation of Input Keys.** The signer obtains (sk, pk) from the security parameter. The forger (i.e. the user) gets as input a public key pk.

2. **Request signature for a Message.** Engage in an interactive signature issue protocol using the honest user algorithm on the message m_0 to get a signature σ_0.

3. **Learn Oracle Queries used in the Protocol.** Let $trans_0$ be the transcript that corresponds to the above protocol execution. Then first run the algorithm $Find_\delta$ to compute the set I_0 that contains the set of intersection queries and then also run the verification algorithm on the signature σ_0 corresponding to m_0.

4. **Guess a Possible Transcript**. Conditioned on the knowledge learned from the previous step, i.e., all query/answer pairs that the user and the verification algorithm made and also all query/answer pairs determined by $Find_\delta$, guess a secret key $\widetilde{sk}$ and an oracle $\widetilde{\mathcal{O}}$ that agree with the information collected about the real secret key sk and oracle $\mathcal{O}$.

5. **Forge a Signature for another Message**. Forge a signature for m_1 using the key $\widetilde{sk}$ and the oracle $\widetilde{\mathcal{O}}$, by running the signature issue protocol locally.

Using the blindness property of the scheme, I'll claim that, the adversary outputs an additional message/ signature pair in the last step with high probability showing that both blindness and unforgeability cannot hold simultaneously for such constructions.

3.3 Notation and Oracles Used

The basic notations are borrowed from the paper [37]. $\mathcal{O}$ is the oracle used, defined as $\mathcal{O} : \pi(0^{n-n'}||f(\cdot))$ where
f : $\{0,1\}^n \rightarrow \{0,1\}^{n'}$, is a many to one function and $\pi : \{0,1\}^n \rightarrow \{0,1\}^n$, is a trapdoor permutation. On input x, the oracle $\mathcal{O}$ outputs the value $\mathcal{O}$(x). Blind signatures with black-box security with respect to an lossy trapdoor permutation oracle $\mathcal{O} : \pi(0^{n-l}||f(\cdot))$ is considered. By $A^{\mathcal{O}}$(x) we mean that an algorithm A on input x gets black-box access to $\mathcal{O}$. Let BS be an oracle

Blind signature scheme. For message 0 (resp. 1), let $trans_0$ (resp. $trans_1$) be a transcript of the execution with $\mathcal{U}(pk, 0)$ (resp. with $\mathcal{U}(pk, 1)$), and let σ_0 (resp. σ_1) be a corresponding signatures. Let $\mathcal{Q}(Vf_0)$ (resp. $\mathcal{Q}(Vf_1)$) denote the set of $\mathcal{O}$ queries made by the verification algorithm Vf $^{\mathcal{O}}(pk, 0, \sigma_0)$ (resp. Vf $^{\mathcal{O}}(pk, 1, \sigma_1)$). Finally, let $\mathcal{Q}(S_0)$ (resp.$\mathcal{Q}(S_1)$) be the set of queries asked by the signer when interacting with $\mathcal{U}(pk, 0)$ (resp. with $\mathcal{U}(pk, 1)$).

3.4 Finding Intersection Queries from the Transcript of a 2-party protocol execution

Before proposing the attacker, we look at the necessary lemma from Barak and Mahmoody-Ghidary [8] . Informally, it states that for any two-party protocol where each party has access to a random oracle there exists an algorithm that, upon observing the transcript of the interaction, finds with high probability all the intersection queries (queries that have been asked by both parties). This result was first discovered by Impagliazzo and Rudich [34] , and a more efficient protocol was given by Barak and Mahmoody-Ghidary [8]. Formally, this result is given in the following lemma.

Lemma 2. *Using [8], we can claim that, if* $\prod$ *be a two-party (randomized) protocol where each party asks at most q oracle queries to the Lossy Trapdoor Permutation Oracle, then for every* $0 < \delta < 1$*, there is an algorithm* $Find_\delta$ *that has access to the messages sent between the 2-party and asks at most* $(10^4 q^2 \backslash \delta^2)$ *oracle queries such that the queries made by* $Find_\delta$ *contain all the intersection queries of the 2 parties with probability at least* $1 - \delta$*.*

We apply this lemma to the scenario of Blind signatures defining the protocol $\prod$ as follows: Corresponding to any oracle Blind signature scheme $BS^{(.)}$, define the following two-party protocol $\prod$ between a signer $\mathcal{S}$ and a user $\mathcal{U}$:

1. S runs $(sk, pk) \leftarrow \mathrm{KG}^{\mathcal{O}}(1^n)$ and sends pk to $\mathcal{U}$.
2. $\mathcal{U}$ and $\mathcal{S}$ then run the signature-issuing protocol on the message 1, at the end of which $\mathcal{U}$ obtains a signature σ_1.
3. $\mathcal{U}$ runs Vf $^{\mathcal{O}}(pk, 1, \sigma_1)$.

Now we'll fix some δ and define $Find_\delta$ (as per Lemma 1) relative to the above protocol $\prod$. Say the above protocol is run in the presence of the trapdoor oracle $\mathcal{O}$. If we let $\mathrm{Q}(\mathcal{S}_\Pi)$ and $\mathrm{Q}(\mathcal{U}_\Pi)$ denote the $\mathcal{O}$-queries made by each party during an execution of the above protocol that resulted in transcript $trans$, then Lemma 1 guarantees that, with high probability, that the set $\mathcal{I}$ contains all the intersection queries, i.e. $\mathrm{Q}(\mathcal{S}_\Pi) \cap \mathrm{Q}(\mathcal{U}_\Pi) \subseteq \mathcal{I}$.

Since the protocol $\prod$ is fixed, we omit this additional input in the following, i.e., $Find_\delta(\text{trans}) := Find_\delta(\prod, \text{trans})$. Note that the message in $\prod$ is fixed, but the transcript might correspond to a different message. Due to the blindness, however, the success probability of the algorithm $Find_\delta$ is independent of the transcript.

3.5 Properties from Blindness

In this section we study the question of what blindness means with respect to the set of intersection queries. The main observation is that due to blindness the set $\mathcal{I}$ that contains all intersection queries must be somehow "independent" of the message. Recall that in the blindness game the semi-honest signer first outputs a public key together with two messages. Then, it interacts with two honest user instances in a random order. The task for the attacker is to predict which user had which message as input. Recall that the algorithm $Find_\delta$, gets as input a transcript (i.e., all the messages exchanged between both parties) of a protocol execution and outputs a set that contains all intersection queries.

Now, consider two protocol executions and suppose that the set of intersection queries depends on the message. Then just by looking at these queries it is possible to determine the order of the messages. To formalize this intuition, consider a (semi-honest) signer $\mathcal{S}^*$ in the blindness game. Since the attacker is semi-honest and by perfect completeness, the user instances get a valid signature. Then, the adversary obtains both signatures in the original order together with the transcripts of both executions.

Now, it is to show that the set of queries that the verification algorithm makes to verify the second message 1 are already contained in the intersection

queries of the first execution.Let's denote $\mathcal{I}_m$, to indicate the set output when $Find_\delta$ is run on the protocol execution $\langle \mathcal{S}(\text{sk}), \mathcal{U}(\text{pk},m_b)\rangle$. In the blindness game, where the message being signed is unknown and $trans_0$ to indicate the transcript of the first protocol execution $\langle \mathcal{S}(\text{sk}), \mathcal{U}(\text{pk},m_b)\rangle$. $trans_1$ is defined similarly for the second execution. Let $\mathcal{I}_0 \leftarrow Find_\delta(trans_0)$ and define $\mathcal{I}_1$ similarly. We also define the sets $\text{Q}(\mathcal{S}_0)$ and $\text{Q}(\mathcal{S}_1)$ similarly for the queries made by the signer.

Now, it's remain to show that, due to blindness all the intersection queries that occur when signing the message 1 are contained in the set $\mathcal{I}_0$. That is, similar to [7], we'll show that 0 is "useful" for 1.

Lemma 3. *Let BS be an oracle Blind signature scheme satisfying blindness. Consider an execution of the blindness experiment, and let Q(KG), Q($\mathcal{S}_b$), $trans_b$, and Q(Vf_b) be as defined above. Then with probability at least* $1-\delta-negl(n)$ *over random coins of the experiment it holds that, Q(Vf_1)* $\cap$ *(Q(KG)* $\cup$ *Q($\mathcal{S}_0$))* $\subseteq$ $Find_\delta(trans_0)$

Proof. We first observe that with probability at least $1-\delta$, we have, $\text{Q}(\text{Vf}_1) \cap (\text{Q}(\text{KG}) \cup \text{Q}(\mathcal{S}_1)) \subseteq Find_\delta(trans_1)$ This follows immediately from Lemma 2 and our definition of protocol π in the previous section.

Now, let's consider now the following adversary $\mathcal{S}^*$:

1. $\mathcal{S}^*$ runs the honest key-generation algorithm to obtain (sk,pk). It records the $\mathcal{O}$-queries Q(KG) made during this step.

2. $\mathcal{S}^*$ then runs the honest signing protocol with the first user instance. Let trans denote the transcript of this execution, and let $\text{Q}(\mathcal{S})$ denote the $\mathcal{O}$-queries made during this step.

3. $\mathcal{S}^*$ then runs the honest signing protocol with the second user instance.

4. $\mathcal{S}^*$ is given signatures σ_0, σ_1 on the messages 0 and 1, respectively. (By perfect completeness, both user instances always obtain valid signatures.) $\mathcal{S}^*$ verifies σ_1 and records the $\mathcal{O}$-queries $\text{Q}(\text{Vf}_1)$ made in doing so.

5. Finally, $\mathcal{S}^*$outputs 1 iff $\text{Q}(\text{Vf}_1) \cap (\text{Q}(\text{KG}) \cup \text{Q}(\text{S})) \subseteq Find_\delta(\text{trans})$.

If b = 1, and so the first user instance represents an interaction with $\mathcal{U}$(pk,1), then $trans = trans_1$ and Q(S) = Q($\mathcal{S}_1$) and so $\mathcal{S}^*$ outputs 1 with probability at least $1 - \delta$. blindness property thus implies that $\mathcal{S}^*$ outputs 1 with probability at least $1 - \delta - negl(n)$ when b = 0 (and the first user instance represents an interaction with $\mathcal{U}$(pk,0)). This concludes the proof. □

3.6 Detailed Proof of the Impossibility Result of Blind Signatures, constructed from Lossy Trapdoor Permutations

In this section, building on the previous definitions and notations, we'll show that there is no black-box construction of Blind signatures from lossy trapdoor functions. To this end, we describe a malicious user $\mathcal{U}^*$ who wins in the unforgeability game when the security of the Blind signature scheme depends on a lossy trapdoor permutation oracle $\mathcal{O} : \pi(0^{n-n'}||f(x))$, where both f is a many to one function and π is a random permutation. This rules out such constructions of Blind signatures.

Like in [37], we also assume that the protocol proceeds in some fixed number of rounds and w.l.o.g. that no party queries $\mathcal{O}$ twice on the same input. We say that any message sent from one party to the other party is a move and assume the signer sends all even moves and the user all odd ones.

Theorem 1. *Let BS be an oracle Blind signature scheme (with perfect completeness) where each party has access to a random oracle. Let q = poly(n) be an upper bound on the number of oracle queries made by KG, $\mathcal{S}$, $\mathcal{U}$, and Vf. Then there exists an adversary which makes at most poly(n) queries and breaks the unforgeability of the scheme with non-negligible probability, where the probability is taken over the randomness of the oracle, key generation, and the randomness of the adversary.*

Proof. To prove this theorem lets first describe the attacker and then analyze its success probability.

Description of the Attacker. Our adversary $\mathcal{U}^*$ works in 5 steps. All the details of the steps are given below :

1. **Setup Algorithm and Key Generation :** The signer generates (sk, pk) from the KG algorithm. The input of the attacker $\mathcal{U}^*$ is a public-key pk. It picks a random values $r_0 \leftarrow \{0,1\}^n$ and $r_1 \leftarrow \{0,1\}^n$.

2. **Signature Issue Protocol as an Honest User.** The adversary $\mathcal{U}^*$ engages in an interactive signing protocol with the external signing oracle on the message 0. $\mathcal{U}^*$ executes the honest user algorithm $\mathcal{U}(pk, 0, r_0)$ obtaining a valid signature σ_0. $\mathcal{U}^*$ then verifies the received signature (observing the oracle queries made by the verification). Let $trans_0$ be the transcript (not including the randomness r_0) of this protocol execution. $\mathcal{U}^*$ then computes the set $\mathcal{I}_0 \leftarrow Find_\delta(trans_0)$. Remember, that by the properties of $Find_\delta$ the set $\mathcal{I}_0$ contains all the intersection queries made by the signer (including key generations) and user (including verification) for this signature. Next, denote by T_0 the complete transcript of the algorithm run so far. i.e., we assume that T_0 contains the secret key sk, public key pk, the message signature pair (0,σ_0), the randomness r_0 of the user and all query-answer pairs made by the key generation Q(KG), the signer Q($\mathcal{S}$), and the user Q($\mathcal{U}$). Note that since the user verifies the generated signature, the set Q($\mathcal{U}$) includes the queries asked by verification Q(Vf(0)). Note further that the attacker $\mathcal{U}^*$ has only partial knowledge of T_0.

3. **Learn Orace Query/Answer Pairs used in the protocol.** Let L_0 be the information that $\mathcal{U}^*$ has about T_0 and the oracle $\mathcal{O}$ following Step 1. This includes : pk, 0, r_0, σ_0, Q($\mathcal{U}_0$) and $\mathcal{I}_0$. Let q be an upper bound on the number of queries asked by each of the BS protocols and let $\delta = 1/10$ be the failure probability of the algorithm $Find_\delta$. Let $\epsilon = \delta/q$ and $M = q/\epsilon\delta$ =100q^2. For $i = 1, ...M$ do the following:

   - Let D_{i-1} be the distribution of T_0, the transcript of the first step, conditioned on only knowing L_{i-1}.
   - Denote by Q(L_{i-1}) the oracle queries that appear in L_{i-1}. If a query $x \in \{0,1\}^n$ / Q(L_{i-1}) appears with probability at least ϵ in D_{i-1}, then $\mathcal{U}^*$ makes this query to $\mathcal{O}$ and adds the query/answer pair to L_i. If there is more than one such query, then he adds the lexicographically first one.

4. **Guess a Possible Transcript.** $\mathcal{U}^*$ samples a random transcript $\widetilde{T}_0$ according to the distribution D_M. Observe that $\widetilde{T}_0$ also defines a secret

key $\widetilde{sk}$ that may be distinct from the real secret key sk. Moreover, $\widetilde{T}_0$ may include some new mappings that were not defined in L_M. These most likely will not match the real oracle $\mathcal{O}$. We let $\widetilde{\mathcal{O}}$ be the following oracle. If a query x appears in $\widetilde{T}_0$ then $\widetilde{\mathcal{O}}$(x) returns the value contained in $\widetilde{T}_0$. Otherwise, $\widetilde{\mathcal{O}}$(x) = $\mathcal{O}$(x).

5. **Forge a Signature for another Message.** To forge a signature, $\mathcal{U}^*$ runs the interactive signing protocol locally using $\widetilde{sk}$ and $\widetilde{\mathcal{O}}$, i.e., $\sigma_1 \leftarrow \langle \mathcal{S}^{\widetilde{\mathcal{O}}}(\widetilde{sk}), \mathcal{U}^{\widetilde{\mathcal{O}}}(\text{pk}, 1; r_1)\rangle$on the message 1. It then verifies the signature σ_1 for the message 1 using the real oracle $\mathcal{O}$. If the signature verifies, then $\mathcal{U}^*$ outputs (σ_0, σ_1) and aborts otherwise.

Analysis of the Attack:

Complexity : $\mathcal{U}^*$ makes at most $poly(n) = M + 10^4 q^2/\delta^2 + O(q)$ oracle queries:

- M for the learning queries step,
- $10^4 q^2/\delta^2$ for running $Find_\delta$, and
- $O(q)$ for generating and verifying the two signatures.

Success Probability : Now, we'll argue that $\mathcal{U}^*$ outputs a successful forgery with probability at least $\frac{4}{5} - \delta - negl(n)$. To analyze the success probability of $\mathcal{U}^*$ let $\widetilde{Q}$(KG), $\widetilde{Q}(\mathcal{S})$ be the queries made by the key generation and the signer during the computation of σ_1. $\widetilde{Q}(\mathcal{U})$ denotes the users queries to $\widetilde{O}$ during the computation of σ_1. Note that the forger only initiated a single protocol execution with the signer but returns two message/signature pairs. Since the forger runs the honest user protocol in the first execution, (0,σ_0) is a valid message/signature pair. Thus, $\mathcal{U}^*$ wins in the unforgeability game as long as Vf $^{\widetilde{\mathcal{O}}}(pk, 1, \sigma_1) = 1$.

In the following we show that, with high probability, the verification algorithm on (1, σ_1) never asks a query on which the oracles $\widetilde{\mathcal{O}}$ and $\mathcal{O}$ disagree. But if the verification algorithm does not ask such a query, it follows by the perfect completeness of the signature scheme that (1, σ_1) must verify as well.

Lemma 4. *Let $Q(Vf_1)$ denote the set of oracle queries made when verifying the signature σ_1. Let $\widetilde{Q}(KG)$ and $\widetilde{Q}(\mathcal{S})$ denote the set of oracle queries made by the key generation and signing algorithms, respectively, in the sampled transcript $\widetilde{T}_0$. Then with probability at least $\frac{4}{5} - \delta - negl(n)$ it holds that, $Q(Vf(1)) \cap (\widetilde{Q}(KG) \cup \widetilde{Q}(\mathcal{S})) \subseteq Find_\delta(trans_0)$*

Clearly, lemma 4 implies our Theorem1. To see this, note that Vf $^{\widetilde{O}}$ (pk, 1, $\sigma 1$) = 1 by perfect completeness of the signature scheme. But the only queries on which $\widetilde{\mathcal{O}}$ and $\mathcal{O}$ can possibly differ are queries in $\widetilde{Q}$(KG) $\cup$ $\widetilde{Q}(\mathcal{S})$ $\setminus$ $Find_\delta(trans_0)$. If verification makes no such queries, then Vf $^{\mathcal{O}}(pk, 1, \sigma_1)$ = Vf $^{\widetilde{\mathcal{O}}}(pk, 1, \sigma_1) = 1$.

Now, let $\mathcal{E}$ denote the event considered in Lemma 4. The proof of Lemma 4 follows the proof in [7] : A series of hybrid distributions are defined where the first hybrid corresponds to the invented transcript $\widetilde{T}_0$ and the transcript of the forgers signature and verification protocols for 1 and the last hybrid corresponds to the transcript produced if all these procedures are executed with respect to the real trapdoor oracle $\mathcal{O}$. The crucial point to look is that, due to the blindness of the signature scheme, event E holds for any pair of messages, and in particular the fixed messages 0 and 1. not only to find any two messages for which event $\mathcal{E}$ occurs by searching through exponentially many (in the number of oracle queries made by the signature scheme) messages. This difference allows the attack to be much more efficient than theirs and in particular, the attack only needs polynomially many oracle queries. Now the Hybrids are defined.

Definition of Hybrid Distributions : We formally define four hybrid distributions $\mathcal{H}_0$, $\mathcal{H}_1$, $\mathcal{H}_2$ and $\mathcal{H}_3$ as follows:

1. **Hybrid $\mathcal{H}_0$:.** The first hybrid is the distribution ($\widetilde{T}_0$, T_1), where $\widetilde{T}_0$ is the invented transcript created by $\mathcal{U}^*$ in Step 4 and T_1 is the transcript of the signature issue and verification protocols for 1 in Step 5. Note that T_0 also includes the queries of the key generation, while T_1 does not.

2. **Hybrid $\mathcal{H}_1$:.** The second hybrid is defined identically to $\mathcal{H}_0$, except that we use $\widetilde{\mathcal{O}}$ to verify the forgers signature σ_1. In $\mathcal{H}_0$, the $\mathcal{O}$ oracle is used instead.

3. **Hybrid $\mathcal{H}_2$:** The third hybrid has the same distribution as $\mathcal{H}_1$, except that we change the definition of $\widetilde{\mathcal{O}}$ as follows. Recall that L_M is the set of $\mathcal{O}$ query/answer pairs that $\mathcal{U}^*$ knows after the learning queries step (Step 3). We define $\widetilde{\mathcal{O}}$ to answer any query contained in L_M with the answer stored there and all other queries x with a random value. This modification results in an oracle $\widetilde{\mathcal{O}}$ that agrees with $\mathcal{O}$ on all the queries $\mathcal{U}^*$ has asked from $\mathcal{O}$ until the end of Step 3 and all the other queries are answered completely at random.

4. **Hybrid $\mathcal{H}_3$:** The distribution of the last hybrid is the same as $\mathcal{H}_2$. except that $\widetilde{T_0}$ is replaced with T_0. Thus the output of this hybrid is (T_0, T_1) which describes the experiment where :

   - The keys are generated $(sk, pk) \leftarrow \mathrm{KG}(1^n)$
   - The signing algorithm uses sk to run $\sigma_0 \leftarrow \langle \mathrm{S}^{\mathcal{O}}(\mathrm{sk}), \mathcal{U}^{\mathcal{O}}(\mathrm{pk},0;\, r_0)\rangle$, and $\sigma_1 \leftarrow \langle \mathrm{S}^{\mathcal{O}}(\mathrm{sk}), \mathcal{U}^{\mathcal{O}}(\mathrm{pk},1;\, r_1)\rangle$.
   - The verification algorithm uses pk to verify both signatures. Note that all algorithms here use the original trapdoor oracle $\mathcal{O}$ and thus verification succeeds for both signatures.

The distributions considered in each hybrid are taken over random choice of the oracle and random coins of the key-generation algorithm, the signer, and the adversary. We prove Lemma 4 by showing that,

1. event $\mathcal{E}$ occurs with high probability in $\mathcal{H}_1$ and
2. the probability that event $\mathcal{E}$ occurs in $\mathcal{H}_0$ is not much smaller than its probability in $\mathcal{H}_3$.

Now, the target is to show that $\mathcal{E}$ occurs with high probability in $\mathcal{H}_3$.

Claim 1: $\Pr_{\mathcal{H}_3}[\mathcal{E}] \geq 1 - \delta - \mathrm{negl(n)}$

This clearly is an immediate consequence of Lemma 2.

Next to show is that the probability of $\mathcal{E}$ remains unchanged when we move from $\mathcal{H}_3$ to $\mathcal{H}_2$.

Claim 2: $\mathcal{H}_2 \equiv \mathcal{H}_3$ Thus, $\Pr_{\mathcal{H}_2}[\mathcal{E}] = \Pr_{\mathcal{H}_3}[\mathcal{E}]$:

Proof. One can view $\mathcal{H}_3$ as being sampled as follows: first, fix L_M; then choose the transcript T_0 at random from D_M. This, however, is exactly the same distribution as $\mathcal{H}_2$ where L_M is fixed and we then choose $\widetilde{T}_0$ from D_M.

Next is to show that $\mathcal{H}_1$ and $\mathcal{H}_2$ are "close".

Claim 3 : $\Pr_{\mathcal{H}_1}[\mathcal{E}] \geq \Pr_{\mathcal{H}_2}[\mathcal{E}] - 1/5$

To prove this, the concept of Statistical Distance, is needed. So, first see what Statistical Distance is.

Statistical distance: If X and Y are two random variables taking values in a finite set A, then Statistical Distance of X,Y denoted by, SD(X,Y) $= 1/2.\ \Sigma_{a \in A}\ |Pr[X = a] - Pr[Y = a]|$.

Now, it is to prove that, SD($\mathcal{H}_1$,$\mathcal{H}_2$) $\leq 1/5$ and hence, $\Pr_{\mathcal{H}1}[\mathcal{E}] \geq \Pr_{\mathcal{H}2}[\mathcal{E}]$ $-1/5$

Proof. Let Q(T_0) be the queries contained in the transcript T_0. Let $\mathcal{B}$ be the event that $\mathcal{U}^*$ ever asks a query in Q(T_0)\Q(L_M). It is clear that $\mathcal{H}_1 = \mathcal{H}_2$ as long as event $\mathcal{B}$ does not occur in either of them, since in both distributions any queries outside of Q(T_0) are answered randomly. This implies that $\Pr_{\mathcal{H}1}[\mathcal{B}] = \Pr_{\mathcal{H}2}\ [\mathcal{B}]$, and SD($\mathcal{H}_1$,$\mathcal{H}_2$) $\leq \Pr_{\mathcal{H}2}\ [\mathcal{B}]$.

Now the task is to show that $\Pr_{\mathcal{H}_2}\ [\mathcal{B}] \leq 1/5$. (In the following, all probabilities are in $\mathcal{H}_2$.) Recall that in Step 2 of the attack, we set $\epsilon = \delta$/q and $\mathcal{U}^*$ learns at most M $= 100q^2$ query/answer pairs from $\mathcal{O}$. Let D_i be the distribution of T_0 sampled in this step by $\mathcal{U}^*$ given the set L_i of known query/answer pairs. Let $\mathcal{C}$ be the event that there are more than M queries that become likely during the attack. That is, $\mathcal{C}$ is the event that there exists a query $x \notin$ Q(L_M) such that x is asked in D_M with probability at least ϵ. Now, one can claim that,

1. $\Pr[\mathcal{C}] \leq \delta = 1/10$ and
2. $\Pr[\mathcal{B}|\neg\mathcal{C}] \leq \delta = 1/10$.

This completes the proof, as,

$\Pr[\mathcal{B}] = \Pr[\mathcal{C}].\ \Pr[\mathcal{B}|\mathcal{C}] + \Pr[\neg\mathcal{C}].\ \Pr[\mathcal{B}|\neg\mathcal{C}]$
$\leq \Pr[\mathcal{C}] + \Pr[\mathcal{B}|\neg\mathcal{C}]$
$\leq 2\delta \ = \ 1/5.$

Now as we've use lossy trapdoor permutation and for two inputs x and $x^{'}$ s.t. $x \neq x^{'}$, it may be possible that $\mathcal{O}(x) \neq \mathcal{O}(x^{'})$ because the function f is a many-to-one function. So, it could well happen that, $f(x) = f(x^{'})$ for $x \neq x^{'}$ due to collision in the function f. So, for a query x_0, asked during signing and verification in $\mathcal{H}_1$, the answer could be same as a guessed answer for a query x_1 in $\widetilde{T_0}$. Similarly, for a query ${x_0}^{'}$, asked during signing and verification in $\mathcal{H}_2$, the answer could be same as a guessed answer for a query ${x_1}^{'}$ in $\mathrm{Q}(T_0)\backslash\mathrm{Q}(L_M)$. So, we don't need any modification in the analysis, that we require in case of blind signature from random permutation. □

The following two claims complete the proof that $\mathcal{H}_1$ and $\mathcal{H}_2$ are close.

Claim 3.1 : Let $\mathcal{C}$ be the event defined in the proof of the previous claim. Then $\Pr_{\mathcal{H}_2}[\mathcal{C}] \leq \delta$.

Proof. All probabilities here are in $\mathcal{H}_2$. Consider an arbitrary query x and let $queried_x$ be the event that x is queried to $\mathcal{O}$ by the signer and then by the user when generating the signature on 0.

Let's assume the following :
$q_x = \Pr[queried_x]$
${Q_x}^{(i)}$ be the event that x is asked in the i^{th} iteration of Step 3.
$p_x(\mathrm{i}) = \Pr[{Q_x}^{(i)}]$ and
$p_x = \Pr[\cup_i\ {Q_x}^{(i)}]$.

Note that,

1. $\sum_x\ q_x \leq q$, since q is an upper bound on the total number of queries asked when running each algorithm of the Blind signature scheme.

2. $\Pr[queried_x|{Q_x}^{(i)}] \geq \epsilon$, since $\mathcal{U}^*$ adds a query to its list only if the probability that this query is asked is at least ϵ.

Hence, $q_x = \Pr[queried_x]$
$\geq \sum_i \Pr[queried_x|{Q_x}^{(i)}].\Pr[{Q_x}^{(i)}]$

$\geq \epsilon.\Sigma_i Pr[\mathrm{Q}_x^{(i)}]$
$= \epsilon.p_x.$

Assume for the sake of contradiction that, $\Pr[\mathcal{C}] > \delta$. Since $\mathcal{C}$ is the event that M queries are learned in Step 2, this implies that the expected number of queries asked, $\sum_x p_x > \delta \mathrm{M}$.
But this would imply, $\delta \mathrm{M} < \sum_x p_x \leq \sum_x q_x/\epsilon \leq \mathrm{q}/\epsilon$, contradicting the fact that $\mathrm{M} = \mathrm{q}/\delta\epsilon$. □

Now, it is left to show the next claim.

Claim 3.2 : Let $\mathcal{B}$ and $\mathcal{C}$ be as defined earlier. Then $\Pr_{\mathcal{H}_2}[\mathrm{B}|\neg\mathcal{C}] \leq \delta$

Proof. Recall that in Step 5, $\mathcal{U}^*$ relies only on the mappings stored in L_M, and all queries from $\mathrm{Q}(T_0)\backslash\mathrm{Q}(L_M)$ are answered at random. But then $\mathcal{H}_2$ is independent of T_0 conditioned on L_M (whereas L_M has the distribution D_M). This means that we can imagine defining $\mathcal{H}_2$ by choosing L_M first, then running $\mathcal{U}^*$ (using L_M) to sample $\mathcal{H}_2$, and then choosing T_0 conditioned on L_M and $\mathcal{H}_2$. Recall that event $\mathcal{C}$ is determined by L_M, and assume that L_M is such that event $\neg\mathcal{C}$ occurs. This implies that every query asked by $\mathcal{U}^*$ that is not in $\mathrm{Q}(L_M)$ must appear in D_M with probability less than ϵ. Since $\mathcal{U}^*$ asks at most q queries in Step 5, the probability that $\mathrm{Q}(T_0)\backslash\mathrm{Q}(L_M)$ contains one of these queries is at most $\epsilon\mathrm{q} = \delta$. □

Finally, we show that E occurs with the same probability in $\mathcal{H}_0$ and $\mathcal{H}_1$.

Claim 4 : $\Pr_{\mathcal{H}_0}[\mathcal{E}] = \Pr_{\mathcal{H}_1}[\mathcal{E}]$

Proof. This claim follows easily if both hybrid distributions $\mathcal{H}_0$ and $\mathcal{H}_1$ use the same oracle $\mathcal{O}$ and if they are sampled using the same random coins for key generation and the adversary (note that the randomness of the adversary fully determines the randomness used to run the honest user algorithm during the signature-issue protocol). But then it follows that event $\mathcal{E}$ occurs in $\mathcal{H}_0$ if and only if it also occurs in $\mathcal{H}_1$. □

This completes the proof of Lemma 4, and thus the proof of Main Theorem. □

Chapter 4

Conclusion

This dissertation thesis is based on Blind Signatures, a special form of digital signature, where the signer remains oblivious about the message, he signs and at the same time, the user can not generate any signature without the help of the signer. The motivation behind constructing blind signatures is mainly its application in Electronic Voting System. With a rapid growth in computer networks, many people can access the network through the Internet and therefore an electronic voting can be a viable alternative for conducting an election. Electronic voting system must attempt to achieve at least the same level of security as ordinary elections that satisfies Confidentiality, Integrity, Authentication and Verifiability as security requirement. Clearly we need blind signature to implement it.

We have surveyed on existing blind signature schemes, and their security. We considered various important blind signature schemes in different models like random oracle model, where a hash function can be seen as an oracle which produce truly random value for each new query, standard model, where no random oracle is used and common reference string model where it is assumed that both parties have access to some string chosen uniformly at random or a string chosen according to some other probability distribution respectively . We also analyzed te various properties used to construct the schemes along with the security proof techniques of the schemes and also the importance of the schemes in the literature of blind signature. We also study different proof techniques of various blind signature schemes. We have also studied blind signatures with some agreed common informations, namely partial blind signature and techniques where the user or the signer may stop

the interactive signature issue protocol prematurely, namely blind signature under aborts. We also considered various prove techniques of impossibility results of blind signatures.

This dissertation thesis considered Lossy Trapdoor Permutations (LTDPs) and their significance in constructing blind signature schemes and proved impossibility of constructing blind signatures from Lossy Trapdoor Permutations (LTDPs) in Random Oracle Model.

Although blind signatures from one way functions and lossy trapdoor functions are ruled out but still the minimal requirement of Blind Signature schemes remain unclear and the most important open question in the context of security of Blind Signature.

Bibliography

[1] M. Abe and J. Camenisch. Partially Blind signatures. *In the 1997 Symposium on Cryptography and Information Security,* 1997.

[2] M. Abe. A Secure Three-Move Blind Signature Scheme for Polynomially-Many Signatures. Eurocrypt 2001.

[3] M. Abe and T. Okamoto. Provably secure partially Blind signatures. In *Crypto 2000, LNCS 1880,* pp. 271- 286. Springer-Verlag, 2000.

[4] N. Asokan, Victor Shoup, and Michael Waidner. Optimistic Fair Exchange of Digital Signatures. Advances in *Cryptology — Eurocrypt'98, Volume 1403 of Lecture Notes in Computer Science,* pages 591-606. Springer-Verlag, 1998.

[5] N. Asokan, Victor Shoup, and Michael Waidner. Optimistic Fair Exchange of Digital Signatures. Advances in *Cryptology – Eurocrypt'98, Volume 1403 of Lecture Notes in Computer Science,* pages 591-606. Springer-Verlag, 1998.

[6] B. Barak, R. Canetti, J.B. Nielsen, and R. Pass. Universally Composable Protocols with Relaxed Set-Up Assumptions. FOCS 2004.

[7] Boaz Barak and Mohammad Mahmoody-Ghidary. Lower bounds on signatures from symmetric primitives. In *48th Annual Symposium on Foundations of Computer Science,* pages 680688, Providence, USA, October 20 23, 2007. IEEE Computer Society Press.

[8] Boaz Barak and Mohammad Mahmoody-Ghidary. Merkle puzzles are optimal - an o(n^2)-query attack on any key exchange from a random oracle. In *Shai Halevi, editor, Advances in Cryptology CRYPTO 2009,*

volume 5677 of Lecture Notes in Computer Science, pages 374390, Santa Barbara, CA, USA, August 16-20, 2009. Springer, Berlin, Germany.

[9] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM CCS* , pp. 6273, 1993.

[10] M. Bellare, C. Namprempre, D. Pointcheval, and M. Semanko. The One-More- RSA-Inversion Problems and the Security of Chaum's Blind Signature Scheme. J. Cryptology 16(3): 185215 (2003).

[11] Rishiraj Bhattacharyya and Avradip Mandal, "On the Impossibility of Instantiating PSS in Standard Model," PKC 2011

[12] A. Boldyreva. Effcient Threshold Signatures, Multisignatures, and Blind Signa- tures Based on the Gap-Diffie-Hellman-Group Signature Scheme. PKC 2003.

[13] Alexandra Boldyreva. Efficient Threshold Signatures, Multisignatures and Blind Signatures Based on the Gap- Diffie-Hellman-Group Signature Scheme. *Public-Key Cryptography (PKC)'03, Volume 2567 of Lecture Notes in Computer Science,* pages 31-46. Springer-Verlag, 2003.

[14] D. Boneh and X. Boyen. Short signatures without random oracles. Journal of Cryptology, 21(2), 149-177, 2008.

[15] S. A. Brands. Untraceable Off-Line Cash in Wallets with Observers. In *Crypto '93, LNCS 773,* pages 302-318. Springer-Verlag, Berlin, 1994.

[16] J. Camensich, M. Koprowski and B. Warinschi. Efficient Blind signatures without random oracles. In Security in *Communication Networks SCN 2004, Springer LNCS 3352,* 134-148, 2004.

[17] J. Camenisch and A. Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In Advances in *Cryptology - CRYPTO 2004, Springer LNCS 3152,* 56-72, 2004.

[18] Jan Camenisch, Gregory Neven, and Abhi Shelat. Simulatable Adaptive Oblivious Transfer. Advances in *Cryp- tology — Eurocrypt'07, Lecture Notes in Computer Science,* pages 573590. Springer-Verlag, 2007.

[19] Ran Canetti, 'Universally Composable Security: A new paradigm for Cryptographic Protocols', 42nd FOCS, 2001. Full version available at http://eprint.iacr.org/2000/067/

[20] R. Canetti, O. Goldreich, and S. Halevi. The random oracle methodology, revisited. In Proceedings of the 13th Annual ACM STOC, pp. 209218, 1998.

[21] Ran Canetti, Eyal Kushilevitz and Yeheuda Lindell, 'On the Limitations of Universally Composable Two- Party Computation Without Set-up Assumptions', Proceedings of EUROCRYPT 2003.

[22] D. Chaum. Blind signature systems. In Advances in Cryptology CRYPTO '83, p. 153. Plenum Press, 1984.

[23] D. Chaum. Blind Signatures for Untraceable Payments. In *Crypto '82,* pages 199-203. Plenum, New York, 1983

[24] D. Chaum, A. Fiat, and M. Naor. Untraceable Electronic Cash. In *Crypto '88, LNCS 403,* pages 319-327. Springer-Verlag, Berlin, 1989.

[25] I. Damgard. Payment Systems and Credential Mechanisms with Provable Security against Abuse by Individuals. Crypto '88.

[26] Seiji Doi, Yoshifumi Manabe , Tatsuaki Okamoto 'Universally Composable Blind Signatures', The 2006 Symposium on Cryptography and Information Security Hiroshima, Japan, Jan. 17-20, 2006

[27] M. Fischlin. Round-optimal composable Blind signatures in the common reference string model. In Advances in *Cryptology-CRYPTO 2006, Springer LNCS 4117,* 60-77, 2006.

[28] N. Ferguson. Extensions of Single Term Coins. *In Crypto '93, LNCS 773,* pages 292-301. Springer- Verlag, Berlin, 1994.

[29] G. Fuchsbauer. Automorphic signatures in bilinear groups and an application to round-optimal Blind signatures. IACR e-print 2009/320. http://eprint.iacr.org/2009/320.

[30] Oded Goldreich. The Foundations of Cryptography, Volume 2. Cambridge University Press, 2004.

[31] J. Groth and A. Sahai. Efficient non-interactive proof systems for bilinear groups. In Advances in Cryptology - EUROCRYPT 2008, Springer LNCS 4965, 415-432, 2008.

[32] L. C. Guillou and J.-J. Quisquater. A practical zero-knowledge protocol fitted to security microprocessor minimizing both transmission and memory. In *C. G. Gunther, editor, Advances in Cryptology – EUROCRYPT '88, volume 330 of Lecture Notes in Computer Science,* pages 123128. Springer-Verlag, 1988.

[33] Carmit Hazay, Jonathan Katz, Chiu-Yuen Koo, and Yehuda Lindell. Concurrently-Secure Blind Signatures Without Random Oracles or Setup Assumptions. *Theory of Cryptography Conference (TCC)'07, Volume 4392 of Lecture Notes in Computer Science,* pages 323-341. Springer-Verlag, 2007.

[34] Russell Impagliazzo and Steven Rudich. Limits on the provable consequences of one-way permutations. In *21st Annual ACM Symposium on Theory of Computing,* pages 44-61, Seattle, Washington, USA, May 15 17, 1989. ACM Press.

[35] Ari Juels, Michael Luby, and Rafail Ostrovsky. Security of Blind digital signatures. In *Advances in Cryptology Crypto97, volume 1294 of Lecture Notes in Computer Science,* pages 150164. Springer-Verlag, 1997.

[36] Y. Kalai, Y. Lindell, and M. Prabhakaran. Concurrent General Composition of Secure Protocols in the Timing Model. STOC 2005.

[37] Jonathan Katz, Dominique Schroder and Arkady Yerukhimovich "Impossibility of Blind Signatures From One-Way Permutations."

[38] A. Kiayias and H.S. Zhou. Concurrent Blind signatures without random oracles. In *Security and Cryptography for Networks SCN06, Springer LNCS 4116,* 4962, 2006.

[39] K. H. Ko, D. H. Choi, M. S. Cho and J. W. Han, "New signature scheme using Conjugality problem," *Cryptology eprint Archieve Report, 2002. (http://eprint.iacr.org/2002/168)*

[40] K. H. Ko, S. J. Lee, J. H. Cheon, J. W. Han, J. S. Kang and C. S. Park, "New public key cryptosystem using braid groups", *Proceedings of Crypto'00, LNCS 1880,* pp. 166-183, 2000.

[41] Y. Lindell. Bounded-Concurrent Secure Two-Party Computation without Setup Assumptions. STOC 2003.

[42] M. Naor. On cryptographic assumptions and challenges. In *Advances in Cryptology - Crypto 2003, Springer LNCS 2729,* 96-109, 200

[43] M.Naor and M.Yung. "Universal One-Way Hash Functions and their Cryptographic Applications". STOC 89

[44] T. Okamoto. Provably secure and practical identification schemes and corresponding signature schemes. In *E. F. Brickell, editor, Advances in Cryptology CRYPTO 92, volume 740 of Lecture Notes in Computer Science,* pages 3153. Springer-Verlag, 1993.

[45] Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. In STOC, pages 187-196, 2008

[46] D. Pointcheval. Strengthened security for Blind signatures. In *K. Nyberg, editor, Advances in Cryptology EUROCRYPT '98, Lecture Notes in Computer Science,* pages 391 405. Springer-Verlag, 1998.

[47] D. Pointcheval and J. Stern. New Blind signatures equivalent to factorization. In *ACM CCS,* pp. 9299. ACM Press, 1997.

[48] D. Pointcheval and J. Stern. Security arguments for digital signatures and Blind signatures. Journal of Cryptology, 13(3):361396, 2000.

[49] D. Pointcheval and J. Stern. Provably secure Blind signature schemes. *In Advances in Cryptology - ASIACRYPT '96. LNCS,* Springer-Verlag, 1996.

[50] G. K. Verma, "Blind signature schemes over braid groups," *Cryptology eprint Archieve Report, 2008. (http://www.eprint.iacr.org/2008/027)*