UNIVERSITY OF WATERLOO

# A Survey on Blind Digital Signatures

**Nabiha Asghar**
Department of Combinatorics & Optimization
University of Waterloo, ON, Canada

This report is a self-contained discourse on the theory of Blind Digital Signatures from 1981 to 2011.

## Table of Contents

# 1. Introduction & Motivation

*We're an information economy. They teach you that in school. What they don't tell you is that it's impossible to move, to live, to operate at any level without leaving traces, bits, seemingly meaningless fragments of personal information. Fragments that can be retrieved, amplified…'*

-- *William Gibson (1981)*

In the 70s and 80s, the telecom world underwent a major technological revolution, with electronic systems taking over most of public and private life domains. Several different e-services emerged for the ease of the customers as well as service providers, such as e-mail, e-shopping, e-payment systems, e-cash, credit cards, fingerprinting, e-voting etc. Since digital networks are not fully secure against hacking, sophisticated cryptographic systems emerged alongside to provide secure transactions over insecure digital networks.

## Digital Signatures

A digital signature is one such cryptographic security scheme which demonstrates the authenticity of a message or a document being transferred from one person to another on an insecure digital network (such as the internet). In other words, it gives the receiver reasons to believe that the message was sent by the claimed sender, thereby providing a way to detect forgery or tampering. The basic scheme, in layman terms, is as follows: the signer S has a private key and a public key. Suppose a user U wants to get a message $m$ signed by S. S uses a signing algorithm, that takes as input $m$ and S's private key, to generate the signature $\sigma$ and sends it back to U. U can verify, using a verification algorithm which uses S's public key, that the signature is valid i.e. it hasn't been forged by someone other than S. Thus, among other things, digital signatures primarily protect against impersonation of parties and, in more serious cases, modification of messages. Note that in a basic digital signature scheme, S gets to see all the messages it signs, and he also knows all the signatures he produces for these messages.

## Need for some privacy

In many applications, there is a need for cryptographic schemes which are similar to digital signatures but not quite the same. U may require blindness i.e. he may want to obtain a signature from S on a secret message $m$ that he does not want to reveal to S. He may also require anonymity/unlinkability i.e. S should not be able to trace the signature back to U. This requires that the signature $\sigma$ must remain unknown to S.

[3] gives a real world example of such a scenario, which may arise when a user wants to spend electronic money at a shop. (An electronic payment scheme is one in which a bank issues electronic money to users who may then spend it at various shops. In such a scheme, the bank may issue electronic money by sending the users signed messages saying 'This is a banknote worth $10. Serial number: S02359593999'). To spend electronic money at a shop, the user will include such an electronic banknote in a payment to the shop. Since these banknotes are trivial to duplicate, however, the shop needs to deposit any received banknotes immediately at the bank to make sure that these banknotes have not been spent already at some other shop. The bank keeps a database containing all spent banknotes, and each banknote can be used for payment only once. If a payment to a shop contains a banknote that was already spent some time before,

the payment is rejected. A basic property of such a payment scheme is that the bank is able to trace exactly at which places a user spends its money. Now a user may desire secrecy about the kinds of items he purchases and the amount of money he spends. In particular, he may want that a particular purchase should not be traceable to him. Notice that these properties automatically hold for purchases done with normal, metal coins. So it is perfectly natural for a customer to desire the same properties in an electronic purchase.

## Blind Signatures

This is where David Chaum [1] entered the arena; he aimed at creating an electronic version of money such that an e-coin could, just like a real 'metal' coin, not be easily traced from the bank to the shop and two spending of the same user could not be linked together. Chaum wanted to mimic these two properties of real coins, namely untraceability and unlinkability, into e-coins. Also, he wanted to make sure that this e-coin is secure against double-spending (i.e. a user should not be able to use the same e-coin more than once; if he does, the bank should be able to figure this out in time). To achieve this goal, he introduced the idea of Blind Signatures in early 80s.

Blind signatures are derivatives of digital signatures and have some additional features. A blind signature scheme is a protocol for obtaining a signature $\sigma$ on $m$ from the signer S such that S does not learn anything about $\sigma$ and $m$. The basic layout of such a protocol is as follows: U generates a secret random number $r$, embeds it into $m$ to obtain $m'$, the masked/blinded message, and sends $m'$ to S. S generates a signature $\sigma'$ on $m'$ and returns it to U. U then removes the random blinding factor to obtain $\sigma$, the signature on $m$. Hence both $m$ and $\sigma$ remain hidden from S. So we see that blind signatures are similar to digital signatures in that they are unforgeable and can be verified against a public key. The difference is that blind signatures are generated by means of a protocol between the signer and a user/requester/receiver such that the signer does not see the message being signed and the signature being generated. Due to this feature, blind signatures are mainly used for electronic cash protocols as a tool to protect customer's privacy.

## Layout of this report

This report aims to be a self-contained discourse on the topic of Blind Signatures. Section 2 gives the definition of a blind signature scheme and its security. Sections 3, 4 and 5 give examples of blind signature schemes based on RSA, discrete logarithm problem (DLP) and the quadratic residue problem respectively. Section 6 talks about the disadvantages of purely blind signatures. Fair blind signatures and partially blind signatures are discussed in sections 7 and 8 respectively, while section 9 gives a brief survey of some other types of blind signatures proposed over the last 30 years. Section 10 talks about the recently published works in the area and the state of the art. We conclude with our final thoughts in section 11.

# 2. Formal Definition of a Blind Signature Scheme

## Scheme Definition

According to [1] and [2], blind signature scheme consists of three parts:

1. Key generation: a probabilistic polynomial time (PPT) algorithm. On input of a security parameter $l$, it outputs a key pair $(sk, pk)$ where $sk$ is the secret key and $pk$ is the public key.
2. Blind signature generation: an interactive and PPT two-party protocol between a signer S and a user U (who receives the signature) with a public key $pk$ as common input. The private input of S is a private key $sk$, and the private input of U is a message $m'$, which is the blinded version of a message $m$. At the end of the protocol, U obtains a either the string '*unsuccessful*' or a signature $\sigma'$ on $m'$ as a private output; S obtains the string '*completed*' or '*not completed*' as private output. U unblinds $\sigma'$ to obtain $\sigma$, the signature on $m$.
3. Blind signature verification: a deterministic polynomial time algorithm. On input of a message $m$, a public key $pk$, and a signature $\sigma$, it determines whether $\sigma$ is a valid signature on $m$ with respect to public key $pk$. If it is valid, the algorithm outputs '*true*', otherwise it outputs '*false*'.

## Security Definitions

According to [1], [2] and [3], a secure blind signature scheme must satisfy the following three properties:

1. Completeness: If the signer and the receiver both comply with the algorithm of blind signature generation, then the blind signature verification algorithm will always accept $\sigma$, the output of the signature generation algorithm (i.e. it will always output '*true*').

2. Blindness: While correctly operating one instance of the blind signature scheme, let the output be $(m, \sigma)$ (i.e. message/signature pair), and the view of the protocol be $V$[1]. At a later time, the signer is not able to link $V$ to $(m, \sigma)$. This essentially means it is infeasible to link any valid pair $(m, \sigma)$ to the instance of the signature generation protocol in which it was created.

3. Unforgeability: A blind signature scheme is unforgeable if for an adversary (who does not know $sk$) the only way to obtain valid pairs $(m, \sigma)$ is to execute the signature generation algorithm with a signer holding private key $sk$. More precisely, a blind signature scheme should withstand a 'one-more forgery' (notion introduced in [9]): if an adversary is able to obtain $k$ valid pairs of messages and signatures, then the signer executed the signature generation protocol at least $k$ times. Preferably, we like this to hold for any positive $k$ bounded polynomially in the security parameter $l$.

**Note: [16] provides formal definitions of unforgeability and blindness in terms of explicitly defined probabilistic algorithms. For a reader who is new to this topic, we stick to semi-formal notions in this section. However, for the sake of completeness, we provide formal definitions of Partially Blind Signatures in section 8. This will serve the purpose of giving the reader a taste of the probabilistic methods underlying the theory of blind signature schemes.*

[1] A view of the protocol consists of all the parameters and values that are visible to the signer (or to any other observer who is monitoring the communication between the signer and the receiver).

Over the past 30 years, many different types of blind signature schemes have emerged that are based on RSA, DLP, Quadratic Residue problem, Bilinear Pairings, Lattices, IDs etc. In addition, the notions of fair-blind signatures, partially-blind signatures and restrictive blind signatures have also been introduced that provide addition security and extra features than a normal blind signature scheme. More recently, quantum blind signatures have emerged and are an active area of research.

We start off by giving examples of some of the early blind signature protocols based on RSA, DLP and Quadratic residue problem.

## 3. Blind signature schemes based on RSA

Chaum ([1], [4], [5]) proposed the first blind signature scheme, which was based on RSA and the hardness of the factoring problem. It is a known fact that e-coins are bit-strings which are vulnerable to being copied and spent more than once. To avoid this double-spending problem, Chaum proposed an RSA-based online e-system where the bank checks online (on-the-go) whether a coin has been spent or not. This was clearly impractical because bank databases are quite large and real time searching takes a lot of time. Chaum then proposed an RSA-based offline version of the blind scheme which tackled the double-spending problem. [7] and [8] give the details of the withdrawal protocols of this e-cash system. The basic protocol is as follows:

Let S be the signer and U be the requester of the signature.

- Key generation: S chooses two random, large primes $p, q$ and computes $n = pq$ and $\phi(n) = (p-1)(q-1)$. Then S chooses two large integers $e, d$ such that $ed \equiv 1\,(mod\,\phi(n))$ and $\gcd\big(e, \phi(n)\big) = 1$. Let $(e, n)$ be S's public key, and let $d$ be S's private key. S keeps $(p, q, d)$ secure and publishes $(e, n)$ and a one-way hash function $H: \{0,1\}^* \rightarrow \mathbb{Z}_n{}^*$ (e.g. SHA-1).

- Blinding: U chooses $r \in_R \mathbb{Z}_n{}^*$ and computes $m' = r^e H(m)\,(mod\,n)$, where $m$ is the message to be signed. U submits $m'$ to S.

- Signing: S computes and sends $\sigma' \equiv m'^{\,d}\,(mod\,n)$ to U.

- Unblinding: U computes $\sigma \equiv \sigma' r^{-1}\,(mod\,n)$.

- Verifying: U verifies the legitimacy of σ on $m$ by checking whether $\sigma^e \equiv H(m)\,(mod\,n)$.

### Security Analysis of Chaum's RSA-based blind signature scheme

The scheme works under the assumption that the RSA problem is infeasible[2].

[2] i.e. there is no PPT algorithm that, given an arbitrary valid RSA public key $(n, e)$ and $y \in_R \mathbb{Z}_n{}^*$, can output $y^d$ $(mod\,n)$ with non-negligible probability.

Completeness: It can be clearly seen that if U and S follow the protocol honestly, then $\sigma^e \equiv (\sigma' r^{-1})^e \equiv (m'^{de})(r^{-e}) \equiv m'^{r^e} \equiv H(m)\ (mod\ n)$. Thus the signature verification algorithm always outputs '*true*'.

Blindness: The use of the random, blinding factor $r$ ensures that the pair $(m, \sigma)$ is statistically independent of the pair $(m', \sigma')$ which can be viewed by the signer during the protocol, thereby implying unlinkability/blindness.

Unforgeability: It is not known whether unforgeability holds for Chaum's blind signature scheme under the standard RSA-infeasibility assumption. [10] shows that one can successfully obtain one-more-forgery if the hash function is poorly implemented. If we assume that $H$ is a random oracle and RSA known-target inversion problem (described in [11]) is hard, it is possible to prove that this scheme is unforgeable. We omit it the long proof here for the lack of space. Interested readers can refer to [11].

Chaum later extended this idea in [12] and obtained schemes similar to RSA-based blind schemes for untraceable coins and untraceable cheques. They ensured customer anonymity if the coin/cheque is used once by the customer, but allowed traceability if he tries to double-spend.

Chaum's RSA-based schemes were quite inefficient, but they laid the foundations for future research in the field. Soon afterwards, more efficient DLP-based blind signatures were presented and consequently, better and efficient e-cash systems were designed.

# 4. Blind signature schemes based on DLP

In 1994, Camenisch *et al*. proposed two DLP-based blind signature schemes in [13]; one was based on the modified Digital Signature Standard (DSA) [14] and the other was based on Nyberg-Rueppel signature scheme [15]. The two schemes are described below.

*Note: The notions of unforgeability were not formalized at the time this work was published, so the security analysis provided here of these two schemes does not include the unforgeability test. These schemes are not currently in use in practical electronic systems and are being provided here for the sake of development of the mathematical theory behind blind signature schemes.*

## Blinded Modified-DSA

- Initialization: The signer S randomly chooses a large prime $p$, a prime factor $q$ of $p-1$, and a generator $g \in \mathbb{Z}_p{}^*$ of order $q$. His secret key is $x \in_R \mathbb{Z}_q$ and the corresponding public key is $y = g^x\ (mod\ p)$. To sign a message $m$ where $\gcd(m, q) = 1$, S additionally selects an integer $\tilde{k}$ at random and computes $\tilde{R} = g^{\tilde{k}}\ (mod\ p)$. S then checks whether $\gcd(\tilde{R}, q) = 1$. If this holds, then S sends $\tilde{R}$ to the requester U.

- Requesting:

1. U checks that $\gcd(\tilde{R}, q) = 1$.

2. U chooses $a, b \in_R \mathbb{Z}_q$ and computes $R = \tilde{R}^a g^b\ (mod\ p)$.

3. U checks whether $\gcd(R, q) = 1$. If not, it goes back to step 2. Otherwise, U computes

$m' = am\tilde{R}R^{-1}\ (mod\ q)$ and sends $m'$ to S.

- Signing:

S computes the signature $\sigma' = \tilde{R}x + \tilde{k}m'(mod\ q)$ and sends it to U.

- Unblinding and Verification:

U unblinds $\sigma'$ by computing $\sigma = \sigma' R\tilde{R}^{-1} + bm\ (mod\ q)$. U also computes $r = R\ (mod\ q)$. The tuple $(r, \sigma)$ is the signature on $m$. U can verify its validity by computing $T = (g^{\sigma}y^{-r})^{m^{-1}}\ (mod\ p)$ and checking that $r = T\ (mod\ q)$.

## Security Analysis of blinded Modified-DSA

Completeness: We need to show that $(r, \sigma)$ is a valid signature on $m$ and for that, it suffices to show that $T = r\ (mod\ q)$. This is easy to show:

$$T = (g^{\sigma}y^{-r})^{m^{-1}} = (g^{\sigma' R\tilde{R}^{-1} + bm}g^{-xr})^{m^{-1}} = (g^{\sigma' R\tilde{R}^{-1} + bm - xr})^{m^{-1}} = (g^{\sigma' R\tilde{R}^{-1} - xr})^{m^{-1}}g^{b}\ mod\ p\ \mathbf{(1)}$$

Now we know that $\tilde{k}a + b = (\sigma' - \tilde{R}x)m'^{-1}a + b = (\sigma' - \tilde{R}x)\left(am\tilde{R}R^{-1}\right)^{-1}a + b\ (mod\ q)$

$$= (\sigma' - \tilde{R}x)\left(m^{-1}R\tilde{R}^{-1}\right) + b = \sigma' m^{-1}R\tilde{R}^{-1} - Rxm^{-1} + b\ (mod\ q)$$

$$= \left(\sigma' R\tilde{R}^{-1} - rx\right)m^{-1} + b\ (mod\ q)$$

Substituting this value of $\left(\sigma' R\tilde{R}^{-1} - rx\right)m^{-1} + b\ (mod\ q)$ in (1) gives us

$$T = g^{\tilde{k}a+b}\ mod\ p = R\ (mod\ p)$$

So $T\ (mod\ q) = R\ (mod\ p)\ (mod\ q) = r\ (mod\ q)$. Hence $T = r\ (mod\ q)$.

Blindness:

Intuitively, it is easy to see that the message-signature pairs $(m, (r, \sigma))$ and $(m', (\tilde{r}, \sigma'))$ , where $\tilde{r} = \tilde{R}\ (mod\ q)$, are statistically independent of each other and hence cannot be linked together due to the random variables $a$ and $b$. However, for the sake of formality and completeness, we give its proof here to give the reader an idea of how to go about proving the blindness property for a signature scheme.

To prove the blindness, it suffices to show that given any view $V$ and a corresponding valid message-signature pair $\left(m, (r, \sigma)\right)$, there exists a unique pair of blinding factors $a$ and $b$. Since the requester U chooses these two numbers randomly and this pair is unique, there is no way that the signer S (or anyone else, for that matter) can guess these two numbers with non-negligible probability. Hence no one can link a view $V$ with its corresponding $(m, (r, \sigma))$ pair. Therefore, the blindness of the scheme follows. Now we show that $a$ and $b$, for a view $V$ (consisting of $\tilde{k}, \tilde{R}, m' and\, \sigma'$) and a valid message-signature pair $(m, (r, \sigma))$ are unique.

Since $(m, (r, \sigma))$ is a valid pair, the following equations must hold (by construction of the scheme):

$m' = am\tilde{R}r^{-1}\,(mod\; q)$ **(1)**

$\sigma = \sigma' r\tilde{R}^{-1} + bm\,(mod\; q)$ **(2)**

$r = \tilde{R}^a g^b\,(mod\; p)(mod\; q)$ **(3)**

Notice that $m$, $\tilde{R}$ and $r = R\,(mod\; q)$ were chosen during the scheme so that they are all relatively prime to $q$. Therefore, $a$ can be uniquely determined by equation (1) and $b$ can uniquely be determined by equation (2):

$$a = m'm^{-1}r\tilde{R}^{-1}(mod\; q), \quad b = \left(\sigma - \sigma' r\tilde{R}^{-1}\right)m^{-1}\,(mod\; q)$$

For blindness to hold, these values of $a$ and $b$, when substituted into (3), should give $r = T\; mod\; q$. This can easily be checked via the following computation:

$\tilde{R}^a g^b = g^{\tilde{k}a+b} = \left(g^{\sigma - xr}\right)^{m^{-1}} = \left(g^{\sigma}y^{-r}\right)^{m^{-1}} = T\,(mod\; q).$ QED

## Blinded Nyberg-Rueppel Signature Scheme

Initialization: The system parameters and initialization here is exactly the same as that in Blinded Modified-DSA scheme.

Requesting: The requester U chooses $a \in_R \mathbb{Z}_q$ and $b \in_R \mathbb{Z}_q{}^*$ and computes $r = mg^a\tilde{R}^b\; mod\; p$ and then computes the blinded message $m' = rb^{-1}(mod\; q)$. U checks if $m' \in_R \mathbb{Z}_q{}^*$. If this is not the case, he chooses some other $a$ and $b$ that satisfy $m' \in_R \mathbb{Z}_q{}^*$. Then U sends $m'$ to the signer S.

Signing: S generates the blind signature $\sigma' = m'x + \tilde{k}\,(mod\; q)$ and sends it to U.

Unblinding & Verification: U unblinds $\sigma'$ by computing $\sigma = (\sigma' b + a)(mod\; q)$. Then the pair $(r, \sigma)$ is the signature on $m$. U can verify the validity of this signature by checking if $m = g^{-\sigma}y^r r\,(mod\; p)$.

## Security Analysis of Blinded Nyberg-Rueppel Signature Scheme

Completeness: We need to show that $(r, \sigma)$ is a valid signature on $m$ and for that it suffices to show that $g^{-\sigma}y^r r = m(mod\; p)$.

. This is easy to show: $g^{-\sigma}y^r r = g^{-\sigma' b - a}g^{rx}\left(mg^a\tilde{R}^b\right) = mg^{-\sigma' b - a + rx + a + \tilde{k}b}\,(mod\; p)$

$$= mg^{-m'xb - \tilde{k}b\, - a + rx + a + \tilde{k}b} = mg^{x(r - m'b)} = mg^0 = m\,(mod\; p)$$

Blindness: This proof is very similar to the blindness proof for the Blinded Modified-DSA described above. To prove the blindness, it suffices to show that given any view $V$ and a corresponding valid message-signature pair $\left(m,(r,\sigma)\right)$, there exists a unique pair of blinding factors $a$ and $b$. Since the requester U chooses these two numbers randomly and this pair is unique, there is no way that the signer S (or anyone else, for that matter) can guess these two numbers with non-negligible probability. Hence no one can link a view $V$ with its corresponding $(m,(r,\sigma))$ pair. Therefore, the blindness of the scheme follows. Now we show that $a$ and $b$, for a view $V$ (consisting of $\tilde{k}, \tilde{R}, m' and\, \sigma'$) and a valid message-signature pair $(m,(r,\sigma))$ are unique.

Since $(m,(r,\sigma))$ is a valid pair, the following equations must hold (by construction of the scheme):

$r = \ mg^a\tilde{R}^b \ mod\ p$ **(1)**

$m' = \ rb^{-1}(mod\ q)$ **(2)**

$\sigma = (\sigma' b + a)(mod\ q)$ **(3)**

Notice that $m'$ was chosen during the scheme so that it is relatively prime to $q$. Due to this, $b$ can be uniquely determined by equation (2) and then $a$ can uniquely be determined by equation (3):

$$b = rm'^{-1}(mod\ q), \quad a = \ \sigma - \sigma' b(mod\ q)$$

As in the previous blindness proof, showing that these values of $a$ and $b$ satisfy equation (1) completes the proof. This is easy to see:

We know from the verification condition that $g^{-\sigma}y^r r = m\ (mod\ p)$ and that $\tilde{R}^b = g^{\tilde{k}b}\ (mod\ p)$. Hence we have

$$\begin{aligned} mg^a\tilde{R}^b = (g^{-\sigma}y^r r)(g^a)\left(g^{\tilde{k}b}\right) &= \ g^{-\sigma+rx+a+\tilde{k}b}r(mod\ p) \\ &= \ g^{-\sigma' b-a+rx+a+\tilde{k}b}r(mod\ p) \text{ by (3)} \\ &= \ g^{-m'xb+rx}r\ (mod\ p) \quad \text{since } \sigma' = m'x + \tilde{k}\ (mod\ q) \\ &= g^{-m'xb+m'bx}r\ (mod\ p) \quad \text{by (2)} \\ &= \ g^0 r\ (mod\ p) = \ r\ (mod\ p). \end{aligned}$$

QED

## Other DLP-based schemes

In addition to the two schemes described above, several other DLP-based schemes have been proposed. [19] proposed a blind signature protocol based on El-Gamal signature scheme. Another commonly used scheme is the blinded version of the Schnorr signature scheme [47] which was proposed by Pointcheval *et al.* in [9]. In [48], an improved version of Okamato-Schnorr blind signature scheme was proposed and it was proved to be secure in the random oracle model as long as the number of issued signatures is logarithmically bounded in the security parameter.

Many of the DLP-based schemes are fair-blind, partially-blind or fair-partially-blind signature schemes. It is worth mentioning here that the most used blind signatures in E-cash systems are based on the discrete logarithm problem and are ‘restrictive’ in nature. (Fair blind signatures have been discussed in Section 7, partially blind signatures in Section 8 and restrictive blind signatures in Section 9).

### Converting any DLP-based digital signature scheme to a blinded signature scheme

In recent past (2006), Qiu [18] has presented some general guidelines that can convert almost any DLP-based signature scheme into a blind signature scheme. He has shown that knowing the process of constructing DLP-based blind signatures and the principle of their use in e-cash systems is valuable for designing new efficient e-cash systems or new blind signatures. The details of his methods are beyond the scope of this paper.

## 5. Blind signature schemes based on Quadratic Residue Problem

### Fan and Lei’s scheme

Fan *et al.* proposed a fast blind signature scheme in [20], based on the quadratic residue problem. This scheme does not require any modular exponentiation or computation of inverses. Instead, a user can obtain a signature and verify it only through modular additions and multiplications, hence the scheme is very efficient. The authors claim that there scheme reduces the number of computations by almost 99%. The protocol is as follows:

Initializing: The signer S randomly selects two large distinct primes $p_1$ and $p_2$, where $p_1 \equiv 3(mod\ 4)$ and $p_2 \equiv 3(mod\ 4)$. S computes $n = p_1 p_2$ and publishes $n$.

Requesting:

- The requester U chooses two random integers $u,v \in \mathbb{Z}_n{}^*$ and computes $a = H(m)(u^2 + v^2) mod\ n$, where $H$ is a hash function such that $H{:}\{0,1\}^* \rightarrow \mathbb{Z}_n{}^*$. Hence $a \in \mathbb{Z}_n{}^*$. This is the blinding step. U sends $a$ to S.
- S selects an integer $x$ such that $a(x^2+1) mod\ n$ is a QR (quadratic residue) in $\mathbb{Z}_n{}^*$. S sends $x$ to U.
- U selects $b\ \in_R \mathbb{Z}_n{}^*$ and computes $d = b^2\ (mod\ n)$ and $\beta = d(ux+v)(mod\ n)$. U sends $\beta$ to S.

Signing: S computes $\lambda = \beta^{-1} \, (mod\, n)$ and computes an integer $t$ such that $t^4 \equiv a(x^2+1)\lambda^2 \, (mod\, n)$. (Note that such a $t$ can exist because both $a(x^2+1)$ and $\lambda^2$ are quadratic residues modulo $n$ and both $p_1$ and $p_2$ are $3\, mod(4)$. Also, S can find this $t$ because he knows the factorization of $n$). S sends the tuple $(t, \lambda)$ to U.

Unblinding and Verification: U unblinds the signature and computes $c = d\lambda(u - vx)(mod\, n)$ and $s = bt(mod\, n)$. The tuple $(c, s)$ denotes the signature of message $m$. One can verify the signature by checking if $s^4 \equiv H(m)(c^2+1)(mod\, n)$.

## Security Analysis of Fan and Lei's Scheme

Completeness: We need to show that $(c, s)$ is a valid signature on $m$ and for that it suffices to show that $s^4 \equiv H(m)(c^2+1)(mod\, n)$. This can be shown as follows:

$$s^4 = b^4 t^4 = b^4 a(x^2+1)\lambda^2 = b^4 H(m)(u^2+v^2)(x^2+1)\lambda^2 = b^4 H(m)\lambda^2[u^2x^2+u^2+v^2x^2+v^2]$$

$$= b^4 H(m)\lambda^2[(u-vx)^2+(ux+v)^2] = b^4 H(m)\lambda^2(u-vx)^2 + b^4 H(m)\lambda^2(ux+v)^2 \;\; mod\, n$$

$$= b^4 H(m)\lambda^2(u-vx)^2 + H(m)[\, b^4\lambda^2(ux+v)^2]\, mod\, n$$

$$= b^4 H(m)\lambda^2(u-vx)^2 + H(m)[\, d^2\beta^{-2}(ux+v)^2]\, mod\, n$$

$$= H(m)d^2\lambda^2(u-vx)^2 + H(m)[\, d^2\beta^{-2}(\beta d^{-1})^2]\, mod\, n$$

$$= H(m)d^2\lambda^2(u-vx)^2 + H(m)\, mod\, n \;\; = H(m)[d^2\lambda^2(u-vx)^2 + 1]\, mod\, n$$

$$= H(m)(c^2 + 1)\, mod\, n$$

Blindness: It can be observed that the integers $b$, $u$ and $v$ are randomly selected therefore the signer (or any viewer of the protocol) cannot link a particular instance of this signing protocol to the signature $(c, s)$ generated at the end.

Unforgeability: Since the factorization of $n$ is unknown to everyone except the signer, a forger who has access to integers $c$ and $m$ can forge a signature $s$ (which is a quadratic residue modulo $n$) only if he can calculate $s$ using the equation $s^4 \equiv H(m)(c^2+1)(mod\, n)$. This involves taking a fourth root modulo $n$, which is hard to do without knowing $p_1$ and $p_2$. Therefore the scheme is unforgeable. Note that this proof holds only under the random oracle model.

## Shao's scheme

Two years after Fan and Lei's scheme was proposed, Shao showed in [21] that:

(i) Fan and Lei's scheme is not as fast as they claimed.
Reason: In the signing-phase, S derives an integer $t$ such that $t^4 \equiv a(x^2+1)\lambda^2 \, (mod\, n)$. Now, since $a(x^2+1)$ and $\lambda^2$ are quadratic residues modulo $n$, this simply means there exists an integer $h$ such that $h^2 \equiv a(x^2+1)\lambda \;\, mod\, n$, and it does not necessarily imply that there exist two more roots of $a(x^2+1)\lambda$, therefore there might not be a fourth root $t$ of $a(x^2 +$

$1)\lambda^2\, mod\, n$. In such a case when no fourth root exists, S would have to go back to the requesting-phase to choose a different value of $x$ and keep going back and forth till a fourth root $t$ can be found. Therefore more computations, and more data transmissions between the signer and requester are required, than the number claimed.

(ii) Fan and Lei’s scheme is not a true blind signature scheme.
Reason: Shao showed that if the requester reveals the signature tuple $(c, s)$, the signer can trace it back to the requester. The reason is that, in an attempt to keep the number of computations low, the requester ends up sending more than necessary information to the signer. The signer can keep a set of records of the 4-tuple $\{a, \beta, x, t\}$ for all the messages he signs, where:
$a = H(m)(u^2 + v^2) mod\, n$,
$\beta = d(ux + v)(mod\, n)$, and

$$t^4 \equiv \frac{H(m)(u^2+v^2)(x^2+1)}{\lambda^2} = \frac{H(m)(u^2+v^2)(x^2+1)}{b^4(ux+v)^2}\ (mod\, n)$$

Suppose now that the requester reveals the signature $(c, s)$, where:
$c = d\lambda(u - vx)(mod\, n) = \frac{u-vx}{ux+v}\ (mod\, n)$, $s = bt(mod\, n)$, and $u, v, b$ are the secret parameters selected by the requester. Then the signer can evaluate the values of these secret parameters through the following equations:

$b = st^{-1}(mod\, n)$ **(1)**

$ux + v = \beta b^{-2}\ (mod\, n) = \beta t^2 s^{-2}(mod\, n)$ **(2)**

$u - vx = c(ux + v) = c\beta t^2 s^{-2}\ (mod\, n)$ **(3)**

From equation (1), $b$ can easily be found. (2) and (3) can then be solved simultaneously to give values of $u\ and\ v$ (note that the inverse of $s$ exists because the inverse of $t$ exists, and the inverse of $t$ exists because $u, v, H(m)$ and $(x^2 + 1)$ all belong to $\mathbb{Z}_n{}^*$. The signer can then check if $a = H(m)(u^2 + v^2) mod\, n$.

Shao then proposed an improved version of this scheme in [21] by fixing these issues and incorporating a few other minor changes. His protocol is as follows:

Initializing: This step is the same as that in Fan and Lei’s scheme.

Requesting: The requester U chooses two random integers $u, b \in \mathbb{Z}_n{}^*$ such that $a = b^2 H(m)(u^2 + 1)\ (mod\, n)$ belongs to $\mathbb{Z}_n{}^*$, where $H$ is a hash function such that $H: \{0,1\}^* \rightarrow \mathbb{Z}_n{}^*$. This is the blinding step. U sends $a$ to the signer S.

Signing: S randomly selects an integer $x$ such that $a(x^2 + 1)\ (mod\, n)$ is a QR in $\mathbb{Z}_n{}^*$. Then S derives an integer $t$ in $\mathbb{Z}_n{}^*$ such that $t^{-2} \equiv a(x^2 + 1)(mod\, n)$. (The signer can do this because he knows the factorization of $n$). S sends the tuple $(t, x)$ to U.

Unblinding and Verification: U computes:

$c = (ux - 1)(x + u)^{-1}(mod\, n)$ **(1)**

$s = bt(x + u)(mod\, n)$ **(2)**

$(c, s)$ is the signature on $m$. To verify the validity of this signature, U checks if

$$H(m)s^2(c^2 + 1) = 1 \ (mod \ n) \qquad \textbf{(3)}$$

### Security Analysis of Shao's Scheme

Shao proved in [21] that this scheme is complete, blind and unforgeable, and that it overcomes the two weakness of Fan and Lei's schemes (we skip the proofs here; they are similar to the proofs for Fan and Lei's technique and are straightforward). Soon afterwards, Fan and Lei published further results in [22] and showed that infact Shao's scheme is vulnerable to Pollard-Schnorr[3] attacks [23]: he proved that an attacker could use this attack to obtain a valid signature $(c, s)$ on a message $m$ without knowing $n$'s factorization, such that the verification condition (3) is satisfied. Hence, Fan and Lei showed that the two schemes offer an almost equivalent level of security as far as blindness is concerned.

## 6. Blind signature schemes – a gateway to perfect E-crimes

Now that we have seen a few examples of blind schemes based on different intractable problems, it is not hard to realize that a tool that provides perfect anonymity to a user can also give him an opportunity to commit a perfect crime. For example, in an e-cash system, even if the concept of blind signatures allows authorities to distinguish between valid and false data (i.e. whether an e-coin has been used previously or not), it prevents the authorities from connecting specific data or actions to specific users.

As Chaum's pioneering work gained popularity in academic as well as industrial circles and began to be improved and implemented in banking systems, Solms *et al.* [24] were the first to point out this grave issue by using a real world crime example and turning it into a perfect crime using blind signatures. To explain their ideas, we first give a brief basic layout of Chaum's e-cash protocol, and then show how it can be taken advantage of by a criminal.

### Simplified Version of Chaum's E-cash protocol ([7] & [12])

Assume the public existence of a one-way function $f$ and an RSA public key $n$ (the factorization of $n$ is kept secret by the bank).

1. A user U requests one unit of electronic money from the bank as follows:

(i) U chooses $r, x \in_R \mathbb{Z}_n{}^*$ and computes $B = r^3 f(x) \ (mod \ n)$. U sends $B$ to the bank.

(ii) The bank computes $D = \sqrt[3]{B} \ (mod \ n)$. Then it withdraws one money-unit from the user's bank account and puts it into a money-pool (in a money-pool, one person's money-unit cannot be distinguished from another person's money-unit). Then the bank sends $D$ to U.

[3] A class of attacks that use the fact that an efficient solution to the congruence $x^2 + ky^2 = m \ (mod \ n)$ can be obtained if k and m are relatively prime to n and if generalized reimann hypothesis is true.

(iii) U computes $C = Dr^{-1} = \sqrt[3]{B}r^{-1} = \sqrt[3]{f(x)}(mod\ n)$. $(x, C)$ represents one legal authorized money-unit that the user can spend. Note that this money unit cannot be traced back to U because it contains $C$, which was computed by multiplying $D$ with a random number known only to U.

2. U spends the money-unit $(x, C)$ as follows:

(i) U offers $(x, C)$ to a shopkeeper in exchange for a good he wants to buy.
(ii) The shopkeeper first verifies the validity of the money-unit by checking if $f(x) = C^3$. If so, he checks with the bank to see if $(x, C)$ has been spent previously. If it has been spent previously, he refuses to accept this money-unit from U. Otherwise he proceeds to the step (iii).
(iii) The shopkeeper gives $(x, C)$ to the bank, and the bank gives him one money-unit from the money pool. Again, there is no way for the bank to trace this money-unit back to U, since it was put in a pool in step 1.

## The Crime

Suppose a person X opens a bank account #19543 at the Bank of Montreal, Canada under a fictitious name and identity John Doe (we assume there are ways of obtaining a fake identity in Canada!). He deposits $10,000 in it and the bank supplies him with  a credit card. About a month later, the baby of a famous Canadian TV actor is kidnapped by a so-called ‘John Doe’ and he threatens to kill the baby if an amount of 1 million dollars isn’t immediately deposited into the bank account #19543. The police get involved in this and quickly find out that John Doe is a fictitious identity. They also figure out that the only way to catch this person now is to keep track of the bank’s ATM operations in real time and hope to arrest the culprit red-handed while he tries to use his credit card to withdraw money at an ATM. The police stations itself near all the main ATMs of the city and a few days later it catches the kidnapper.

## The perfect crime

Using the idea of blind signatures, X could have committed a perfect crime as follows:

1. Open the bank account, receive the credit card and kidnap the baby.

2. Choose a set of $x'$s (i.e. $x_1, x_2, x_3, \ldots., x_p$) and a set of $r'$s (i.e. $r_1, r_2, r_3, \ldots., r_p$).

3. For all $j \in \{1, \ldots, p\}$, compute $B_j = r_j{}^3 f(x_j)\ (mod\ n)$ and mail the set $\{B_1, B_2, \ldots., B_p\}$ to the authorities with the threat to kill the baby if the following instructions are not complied with:

- For all $j \in \{1, \ldots, p\}$, compute $D_j = \sqrt[3]{B_j}\ (mod\ n)$
- Publish the set $\{D_1, D_2, \ldots., D_p\}$ in a newspaper.

4. Buy the newspaper after the set $\{D_1, D_2, \ldots., D_p\}$ has been published. For all $j \in \{1, \ldots, p\}$, compute $C_j = D_j r^{-1}\ (mod\ n)$. Now $(x_j, C_j)$ represents one legal and authorized money-unit for each $j$.

### Analysis of the Perfect Crime

Notice the similarity between the form of the money-unit derived in Chaum's protocol and the one derived in the scenario of the perfect crime. For all $j \in \{1, \dots, p\}$, $C_j$ cannot be traced back to U because no one can find out if it has been derived from one of the $D_j$s published in the newspaper. U can deposit these money-units into his account at an ATM now, without being afraid of getting caught because no one suspects these money-units and no one suspects his account number.

Keeping this pitfall of blind signatures in sight, a new variation called 'fair blind signatures' was proposed by Stadler *et al*. [25]. (He got inspiration from the concept of fair cryptosystems, introduced by Micali [26] some time earlier).

## 7. Fair Blind signature schemes

A fair blind signature is a blind signature with revocable anonymity and unlinkability due to the involvement of a trusted third-party.

Fair blind signature schemes have the addition property that a trusted third-party is involved, which possesses certain information that can help link a signer's view of the protocol to the message-signature pair, in case of a fraud or dishonest transaction such as blackmailing (as described in the example above) and money laundering. It is useful if the anonymity of a user could be removed with the help of this trusted third party, when this is required for legal reasons.

### The model and its types

The model of a fair blind signature scheme primarily consists of a sender (possibly more than one), a signer, a trusted third-party (e.g. a judge), and two protocols:

- a signing protocol involving the signer and a sender
- a link-recovery protocol involving the signer and the judge

The signing protocol functions exactly the same way as that in any ordinary blind signature scheme (e.g. the ones described in previous sections). The link-recovery protocol allows a signer to get information from the judge that enables him to link a view $V$ of the protocol to its corresponding message-signature pair.

There are two types of fair blind signature schemes, depending on the information the judge receives from the signer during the link-recovery protocol:

*Type 1:* Given the signer's view of the protocol, the judge delivers information that enables the signer (or anyone else) to efficiently recognize the corresponding message-signature pair (i.e. the judge can extract the message $m$ that was signed).

*Type 2:* Given the message-signature pair, the judge delivers information that enables the signer to efficiently identify the sender of that message or to find the corresponding view $V$ of the signing protocol.

The model can be visually summarized as follows:

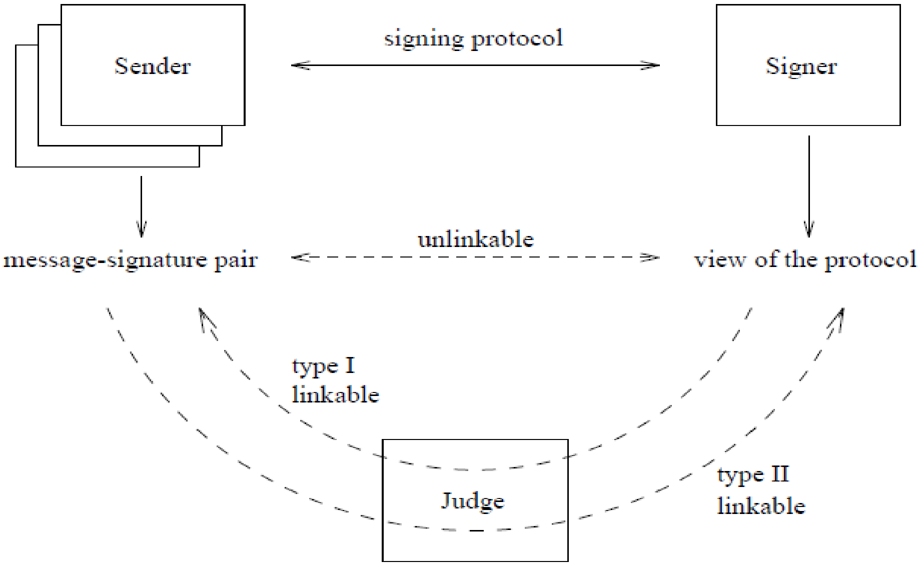
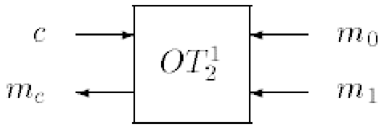


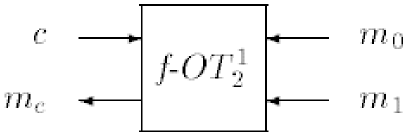
Figure: The model for a fair blind signature scheme. (Screenshot: [25] pp. 3)

We present one of Stadler *et al.*'s schemes here to illustrate how the concept can be implemented. Their schemes were inefficient to some extent because of large amount of data transactions involved due to the involvement of another party in the protocol. However, his techniques paved way for the production of more efficient and secure schemes later on.

Definition 1: A One-out-of-two Oblivious Transfer Protocol (denoted by $OT_2^1$ ) is a protocol between a sender and a receiver which allows the receiver to choose one of the two messages sent by the sender in a way that he receives only the chosen message and the sender does not know which message the receiver has chosen. Let $m_0$ and $m_1$ denote the two messages sent by the sender and let $c$ be the selection bit of the receiver. Then an execution of the $OT_2^1$ protocol is denoted by:



Definition 2: A *fair* one-out-of-two oblivious transfer protocol (denoted by *fair*-$OT_2^1$ ) is a modification of $OT_2^1$ that allows a judge, but not the sender, to determine the selection bit $c$. An execution of the *fair*-$OT_2^1$ protocol is denoted by:



(Stadler *et al.* gave an implementation of a *fair*-$OT_2^1$ in [25], which is beyond the scope of this paper. We simply assume that such a protocol exists). Using this protocol, Stadler *et al.* converted a variation of the Fiat-Shamir-Signature Scheme [27] into a fair blind signature scheme.

## A fair blind signature scheme based on a variation of Fiat Shamir Signature Scheme [27]

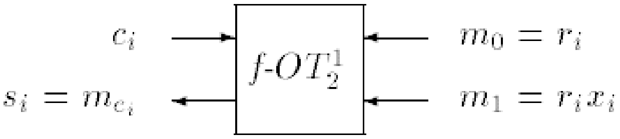
System Parameters: Let $n = pq$ be the product of two large primes chosen by the signer such that $\gcd\big(3, \phi(n)\big) = 1$. Let $y \in_R \mathbb{Z}_n{}^*$. Further, let $H$ denote a one-way hash function and $k$ be a security parameter (e.g. $k > 80$). Define the sequences:

$$y_i = H(y+i)(mod\, n), \qquad x_i = y_i{}^{\frac{1}{3}}\,(mod\, n), \qquad i = 1, \dots, k$$

Initializing: The pair $(n, y)$ is the signer S's public key. Note that only S can compute the sequence $x_i$; he is the only one who knows the factorization of $n$, therefore only he can compute the third roots of $y_i$ modulo $n$. S chooses $r_1, r_2, \dots, r_k \in_R \mathbb{Z}_n{}^*$, and computes $t = \prod_{i=1}^{k} r_i{}^3\ (mod\, n)$. S sends $t$ to the requester U.

Requesting & Signing:

1. U chooses $a \in_R \mathbb{Z}_n{}^*$, and computes:
   - $\tilde{t} = ta^3(mod\, n)$
   - $c = H(\tilde{t}||m)$, where || denotes concatenation of strings. Let $c_i$ be the $i$'th bit of $c$.
2. U and S now engage in a *fair*-$OT_2^1$ $k$ times:
   For $i = 1, \dots., k$:
   - U submits a bit $c_i$ to *fair*-$OT_2^1$.
   - S submits two messages $m_0 = r_i$ and $m_1 = r_i x_i$ to *fair*-$OT_2^1$.
   - U receives $s_i = m_{c_i}$.



3. U computes $\tilde{s} = a \prod_{i=1}^{k} s_i\ (mod\, n)$. The pair $(\tilde{s}, \tilde{t})$ is a signature on $m$.

Verifying:

U checks if $\tilde{s}^3 = \tilde{t} \prod_{i=1}^{k} y_i{}^{c_i} \ (mod\ n)$ holds.

## Security Analysis of the Fair-Blind Variation of Fiat Shamir Scheme

<u>Completeness:</u> We show that if S and U follow the protocol honestly, then $\tilde{s}^3 = \tilde{t} \prod_{i=1}^{k} y_i{}^{c_i} \ (mod\ n)$ holds.

Note that $m_{c_i}{}^3 = \begin{cases} r_i{}^3 = r_i{}^3 H(y+i)^{c_i}, & if\ c_i = 0 \\ (r_i x_i)^3 = r_i{}^3 y_i = r_i{}^3 y_i{}^{c_i} = r_i{}^3 H(y+i)^{c_i}, & if\ c_i = 1 \end{cases}$

Hence $m_{c_i}{}^3 = r_i{}^3 H(y+i)^{c_i}$.

Therefore $\tilde{s}^3 = a^3 \prod_{i=1}^{k} s_i{}^3 = a^3 \prod_{i=1}^{k} m_{c_i}{}^3$

$$= a^3 \prod_{i=1}^{k} r_i{}^3 H(y+i)^{c_i} = a^3 \prod_{i=1}^{k} r_i{}^3 \prod_{i=1}^{k} H(y+i)^{c_i}$$

$$= ta^3 \prod_{i=1}^{k} H(y+i)^{c_i}$$

$$= \tilde{t} \prod_{i=1}^{k} y_i{}^{c_i} \ (mod\ n)$$

<u>Blindness:</u>

Clearly, after the protocol has ended, there is no way for the signer to know what $\tilde{s}\ and\ \tilde{t}$ are because each is a multiple of $a$, a value known only to the requester U. Similarly $m$ remains hidden from S. So we need to show that if the signature pair $(\tilde{s}, \tilde{t})$ is revealed to S later, he cannot link it to $V$, the view of the protocol.

Note that $\tilde{s}$ does not give S any information about $s_i$. Also, by the assumptions of *fair*-$OT_2^1$, the signer S cannot determine the selection bits $c_i$ since only the judge knows them. Therefore, S cannot determine whether $m_0$ or $m_1$ was chosen in any iteration of the protocol. So, $\tilde{s}$ does not give S any information about $V$.

It remains to show that $\tilde{t}$ is also equally unhelpful for S. $\tilde{t}$ can be linked to $V$ only if $\tilde{t}$ can be linked to $t$. But this is not possible because there is only a unique value of $a$ that satisfies $\tilde{t} = ta^3 (mod\ n)$, and this value is not known to S. So it can be concluded that the signature is independent of $V$ and the scheme is perfectly blind.

<u>Fairness:</u> If the sender sends $V$ to the judge, the selection bits $c_i$, and consequently $c$ can be determined. This value of $c$ could be flagged, so that anyone, particularly the signer, can recognize this message-signature pair later.

## Other fair blind signature schemes

Several fair blind signature schemes have been proposed after this, such as [28], [29], [30] and most recently, [31].

[28] proposed an efficient and fair offline e-cash system based on the idea of fair blind signatures. In [29], Abe et al. pointed out that it was important to study a fair blind scheme's security in terms of revocability, in addition to unforgeability. They proposed an efficient fair blind scheme with a complete security analysis, which shows that their scheme is secure against most adaptive and parallel attacks; it claims to offer one-more unforgeability and 'tight' revocation i.e. given a signature, revocation identifies the issuing session that uniquely produced the signature, and, given a session view, revocation identifies the unique signature created in the session. [29] is one of the first few works which discussed the security of a fair blind signature scheme in full detail. Their scheme is too lengthy to be described here, so we omit it.

In [30], Hufschmitt and Traore pointed out a flaw in the proof of unforgeability of [29] and proposed a stronger security model for fair blind signatures than theirs. They also presented a new fair blind scheme based on bilinear maps that satisfies it in the random oracle model under an interactive assumption.

[31] will be discussed in Section 10.

# 8. Partially Blind signature schemes

## Motivation[4]

Typically, a digital signature comes with not just the document body but also attributes such as 'date of issue' or 'valid until', which may be controlled by the signer rather than the receiver. One particular shortcoming of fully blind signatures is that, since the signer's view is perfectly shut off from the resulting signatures, the signer has no control over the attributes except for those bound by the public key. An example is, 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 can seriously effect performance. A similar shortcoming can be seen in a simple e-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.

Partially blind signatures were first introduced in [32]; they are a generalized notion of blind signatures. A partially blind signature scheme allows the signer to explicitly include common 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. By fixing common information to a single string, one can easily transform partially blind signature schemes into fully blind ones. However, the reverse is not that easy.

Partially blind signatures can be regarded as ones lying between ordinary non-blind digital signatures and fully blind signatures, so they should satisfy the security requirements assigned to ordinary digital signatures and those of blind signatures.

[4] Taken from [33]

## Formal Definition of a partially blind signature

[33] describes a partially blind signature scheme as follows: In the scenario of issuing a partially blind signature, the signer and the receiver 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 receiver to the signer. This negotiation is done outside of the signature scheme, and the signature scheme is to be secure regardless of the process of agreement. This notion is formalized by introducing function $Ag(\,)$ which is defined outside of the scheme. Function $Ag$ is a polynomial-time deterministic algorithm that takes two arbitrary strings $info_s$ and $info_u$ that belong to the signer and the user (receiver), respectively, and outputs $info$. To compute $Ag$, the signer and the user exchange $info_s$ and $info_u$ with each other. (Note: 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 $info_u$.)

*Note: Now that the reader is familiar with the working of blind signatures, we give the formal definition of a partially blind scheme below, as promised.*

Formally, a partially blind scheme consists of:

1. Key generation: a PPT algorithm $\mathcal{G}$. On input of a security parameter $l$, it outputs a key pair $(sk, pk)$ where $sk$ is the secret key and $pk$ is the public key.
2. Partially-blind-signature generation: an interactive and PPT two-party protocol between two algorithms, $\mathcal{S}$ (which represents the signer) and $\mathcal{U}$ (which represents the user). The public input of $\mathcal{U}$ is $pk$, $info_u$ and description of $Ag$. The public input of $\mathcal{S}$ is the description of $Ag$ and $info_s$. The private input of $\mathcal{S}$ is a private key $sk$, and the private input of $\mathcal{U}$ is a message $m$. At the end of the protocol, $\mathcal{U}$ either obtains the string '*unsuccessful*' or the tuple ($info$, $m$, $\sigma$) as a private output, where $\sigma$ is the signature on $m$; $\mathcal{S}$ obtains the string '*completed*' or '*not completed*' as private input.
3. Partially-blind-signature verification: a deterministic polynomial time algorithm $V$. On input of $info$, message $m$, a public key $pk$ and a signature $\sigma$, it determines whether $\sigma$ is a valid signature on $m$ with respect to public key $pk$ and common information $info$. If it is valid, the algorithm outputs '*true*', otherwise it outputs '*false*'. (i.e. V($info, m, pk, \sigma$) = '*true*' or '*false*'.)

A secure partial blind scheme has the following three properties:

1. Completeness: If the signer and the user both comply with the algorithm of blind signature generation, then the private output of $\mathcal{S}$ is '*completed*', the private output of $\mathcal{U}$ is ($info$, $m$, $\sigma$), $info = Ag(info_s, info_u)$, and the verification algorithm outputs '*true*'.

2. Partial blindness:

Let $U_0$ and $U_1$ be two honest users that follow the signature issuing protocol. Let $S^*$ be a PPT algorithm that denotes a signer and let $(sk, pk)$ be a key pair generated by the key generation algorithm.

(a) Let $(msg_0, msg_1, info_{u0}, info_{u1}, Ag) \leftarrow S^*(sk)$.

(b) Set up the input tapes of $U_0$, $U_1$ as follows:

- Select $b \in_R \{0,1\}$ and put $msg_b$ and $msg_{\bar{b}}$ on the private input tapes of $U_0$ and $U_1$, respectively ($\bar{b}$ denotes $1 - b$)

- Put $info_{u0}$ and $info_{u1}$ on the public input tapes of $U_0$ and $U_1$, respectively.
- Also put $pk$ and $Ag$ on their public input tapes.

(c) (c)$S^*$ engages in the signature protocol with $U_0$ and $U_1$.
(d) If $U_0$ and $U_1$ output $(info_{u0}, msg_0, \sigma_b)$ and $(info_{u1}, msg_1, \sigma_b)$, respectively, on their private tapes, and $info_{u0} = info_{u1}$ holds, then give those outputs to $S^*$. Give '*unsuccessful*' to $S^*$ otherwise.
(e) $S^*$outputs $b' \in \{0,1\}$.

We say that $S^*$ wins if $b' = b$.

A signature scheme is partially blind if any PPT algorithm $S^*$ wins with probability which is at-most negligibly more than $\frac{1}{2}$. Note that this definition is similar to the definition of blindness that we gave in section 2. The only difference is that now there is an additional factor $info$ that needs to be accounted for. This essentially means it is infeasible to link any valid signature-tuple to the instance of the signature generation protocol in which it was created.

3. Unforgeability:

There may be two possible ways of forging a partially blind signature: the forger could either

(i) forge $info$, or
(ii) for a fixed $info$, he could produce $k_{info} + 1$ signatures, given only $k_{info}$ successful executions of the signature-generation-protocol (note: here $k$ depends on $info$, hence the notation $k_{info}$).

One can observe, however, that (i) is the same as (ii), when $k_{info} = 0$ (i.e. if the signature-generation-protocol has never been run for a particular $info$, a forger can produce a signature only if he can forge $info$). So essentially, a forgery means successfully accomplishing (ii).

To define this notion of unforgeability formally, we define the following 'game':

Let $S$ be an honest signer who follows the signature issuing protocol. Let $U^*$ be a PPT algorithm that denotes a forger and let $(sk, pk)$ be a key pair generated by the key generation algorithm.

(a) Let $Ag \leftarrow U^*(pk)$.
(b) Put $sk$, $Ag$ and a randomly taken $info_s$ on the proper tapes of $S$.
(c) $U^*$ engages in the signature-generation-protocol with $S$. For each $info$, let $k_{info}$ be the number of executions of the signature generating protocol where S outputs '*completed*' with $info$ on its output tape. (if an $info$ never appears on the output tape, let $k_{info} = 0$).
(d) $U^*$outputs one single $info$, and $k_{info} + 1$ message-signature pairs $((m_0, \sigma_0), (m_1, \sigma_1), \dots (m_{k_{info}+1}, \sigma_{k_{info}+1})$.

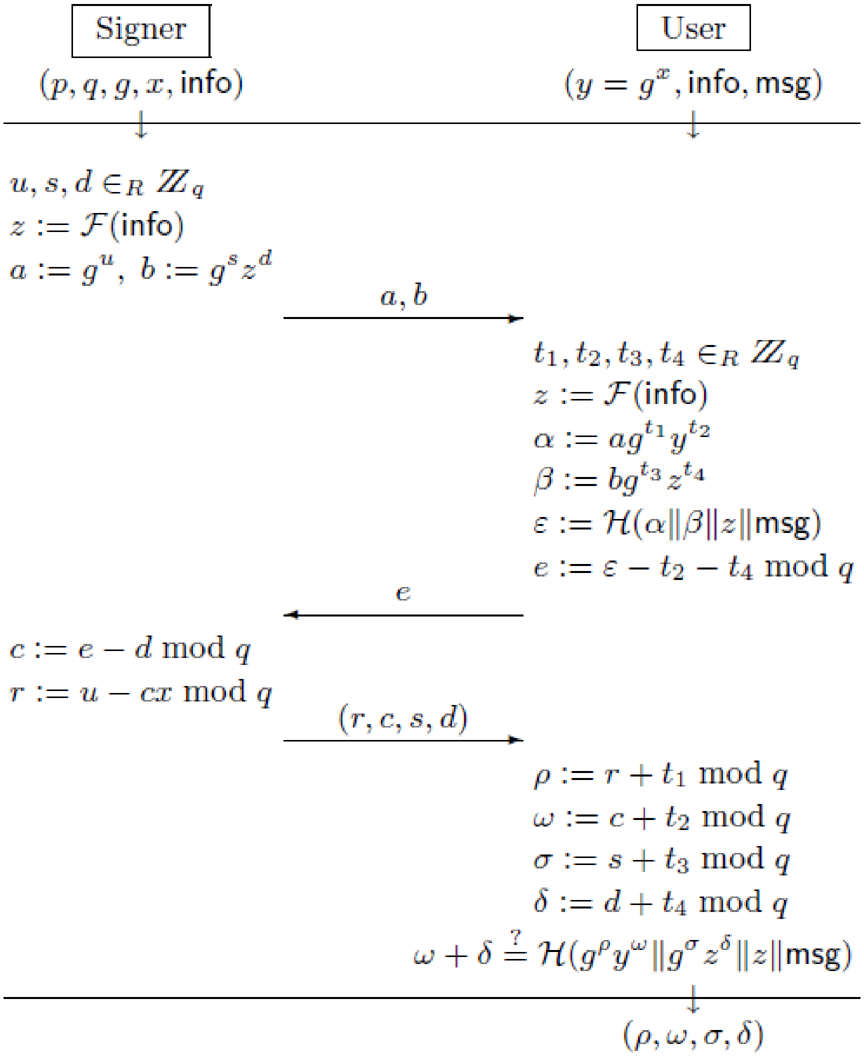
We call a partially blind scheme unforgeable if the probability of the output of $U^*$ satisfying V$(info, pk, m_j, \sigma_j)$ = '*true*' for all the $j = 0, 1, \dots, k_{info} + 1$ is negligible.

## Partially Blind WI-Schnorr Signature Scheme

This is a DLP based partially-blind scheme, the first of its kind proposed by [33], together with a full security definition and proof for the very first time.

Let $p, q$ be large primes such that $q|p-1$. Let $g \in \mathbb{Z}_p{}^*$ such that $g$ has order $q$. Let $\langle g \rangle$ denote the subgroup of $\mathbb{Z}_p{}^*$ of order $q$. Let $H: \{0,1\}^* \to \mathbb{Z}_p$ and $F: \{0,1\}^* \to \langle g \rangle$ be public hash functions. Let $x \in \mathbb{Z}_q$ be a secret key and $y = g^x (mod\ p)$ be the corresponding public key.

The signer $S$ and receiver $U$ first agree on the common information $info$ in a pre-determined way. Then they execute the following partially blind signature-issuing protocol (all the arithmetic operations are done modulo $p$):



Signer

$(p, q, g, x, \mathsf{info})$

User

$(y = g^x, \mathsf{info}, \mathsf{msg})$

$u, s, d \in_R \mathbb{Z}_q$

$z := \mathcal{F}(\mathsf{info})$

$a := g^u,\ b := g^s z^d$

$\xrightarrow{a, b}$

$t_1, t_2, t_3, t_4 \in_R \mathbb{Z}_q$

$z := \mathcal{F}(\mathsf{info})$

$\alpha := a g^{t_1} y^{t_2}$

$\beta := b g^{t_3} z^{t_4}$

$\varepsilon := \mathcal{H}(\alpha \| \beta \| z \| \mathsf{msg})$

$e := \varepsilon - t_2 - t_4 \bmod q$

$\xleftarrow{e}$

$c := e - d \bmod q$

$r := u - cx \bmod q$

$\xrightarrow{(r, c, s, d)}$

$\rho := r + t_1 \bmod q$

$\omega := c + t_2 \bmod q$

$\sigma := s + t_3 \bmod q$

$\delta := d + t_4 \bmod q$

$\omega + \delta \overset{?}{=} \mathcal{H}(g^\rho y^\omega \| g^\sigma z^\delta \| z \| \mathsf{msg})$

$(\rho, \omega, \sigma, \delta)$

Figure: The partially blind WI-Schnorr signature issuing protocol (Screenshot: [33] pp. 7)

The resulting signature for message $msg$ and common information $info$ is a four-tuple $(\rho, \omega, \sigma, \delta)$. A signature is valid if it satisfies

$$\omega + \delta \equiv H(g^\rho y^\omega \| g^\sigma F(info)^\delta \| F(info) \| msg)(mod\ q)$$

### Security Analysis of Partially Blind WI-Schnorr Signature Scheme

[33] gives an in-depth proof of this scheme's completeness, partial blindness and unforgeability in the random oracle model, assuming the hardness of DLP. The proof of completeness goes along the same lines as in the schemes outlined previously. The proofs of the remaining two properties are long so we omit them here due to brevity of space. However, the key point to note is that this scheme is *partially* blind because the common information $info$ is known to both the user and the signer and it is embedded into the signature. Hence, only a part of the signature is blind for the signer, and this serves many purposes in practical applications involving blind signature protocols.

### Some other partially blind schemes

In 2004, [34] came up with the proposal for partially blind signature schemes based on bilinear pairings that give signatures of short size. The article gives the proof of their security in the random oracle model.

More recently, in 2010, [35] presented a partially blind signature scheme, based on the Schnorr signature scheme [48], that has been attractive for mobile clients and smart-card implementation in e-commerce applications due to its low computation-level. It uses this scheme to propose a fair e-payment protocol that does not require the intervention of the third party. The basic idea provided by this article can be used with any partially blind signature scheme to construct new fair e-payment protocols of high efficiency.

## 9. Other types of blind signature schemes

So far we have described the main types of blind signatures, namely normal, fair and partially blind signature schemes. There are some other variants that have been proposed by various researchers over time. We just mention them here briefly and give references for the interested readers.

1. The concept of restrictive blind signatures was proposed by Brands [17] and has played a very important role in the area of e-cash for the last 15 years. The idea is that it allows a recipient to receive a blind signature on a message not known to the signer, but the choice of the message is restricted and must conform to certain rules. Brand proposed a highly efficient e-cash system, where the bank ensures that the user is restricted to embed his identity in the resulting blind signature. When spending the coin at a merchant, the user proves to the merchant her knowledge on the 'inside' construction using the zero knowledge proof. When double spending the coin, two points of a 'line' in the zero knowledge proof will be exposed, and the coefficients of the line can then be computed and used to reveal the 'inside' construction of message. Consequently, knowing the 'inside' construction results in revealing the identity information of the user.
2. 'Magic Ink' signatures were proposed in [36] to solve the perfect crime resulting from unconditional anonymity provided by normal blind signatures.
3. In a proxy blind signature ([37] and [38]), the signer delegates his/her signing power to a proxy, who blindly signs a message on behalf of the original signer.
4. In [39] and [40], forward-secure blind signature scheme were proposed to address key exposure problem, in which all previously generated signatures are still considered to be valid even if the secret key is compromised. They give an extra level of security to normal blind signature.

5. Group[5] blind signatures were proposed in [41].
6. In [42] and [43], blind threshold signatures were devised that enable any $t$ out of $n$ legitimate signers to give a blind signature.
7. A blind threshold ring[6] signature, providing signer-ambiguity, was considered in [44].
8. Blind multi-signatures were proposed in [45].
9. In [49], Galindo *et al*. proposed generic constructions of identity-based signature schemes with additional properties such as identity-based blind (or partially blind) signatures from PKI[7]-based signature schemes with the same properties. Identity-based blind signatures will be discussed in the next section.

# 10. State of the art

Since Chaum's first proposal of the idea of blind signatures in early 80s, it has become a cornerstone in privacy-oriented cryptography e.g. anonymous internet banking, e-voting and oblivious transfer systems. With emerging technologies, however, there is a constant threat of new attacks. While factoring, DLP and quadratic residue problems are still considered to be intractable, the question is whether they will remain so in (near) future. With the advent of quantum computing, for example, there can potentially be a huge leap in computing power (for instance, the *Shor's Algorithm* for quantum computers can solve integer factorization and DLP in polynomial time!). Most of the problems considered intractable right now have been conjectured to be tractable in post-quantum age. To cater to this issue, there is a need to look for problems that can withstand quantum computer attacks (or any other class of attacks that may come to light in future). Most of the current research in this area is geared towards:

(i) creating schemes based on the hardest possible problems (such as lattices), and
(ii) coming up with proofs of security that rely on the mildest possible assumptions (such as bypassing the use of the random oracle model).

In this section, we give brief summaries of recently published works that seem promising. We then delineate some of their shortcomings and suggest future work that can be done to improve the results.

## Lattice-based Blind Signatures

It has been conjectured that the hardness of finding short vectors in a lattice is *post-quantum* (i.e. it can withstand quantum computer attacks). Unlike factoring, even today the hard lattice problems can withstand sub-exponential attacks and the best known algorithms have an exponential complexity in the

[5] Wikipedia: A **Group signature scheme** is a method for allowing a member of a group to anonymously sign a message on behalf of the group. One application is keycard access to restricted areas where it is inappropriate to track individual employee's movements, but necessary to secure areas to only employees in the group.
[6] Wikipedia: "a **ring signature** is a type of digital signature that can be performed by any member of a group of users that each have keys. Therefore, a message signed with a ring signature is endorsed by someone in a particular group of people. One of the security properties of a ring signature is that it should be difficult to determine *which* of the group members' keys was used to produce the signature. Ring signatures are similar to group signatures but differ in two key ways: first, there is no way to revoke the anonymity of an individual signature, and second, any group of users can be used as a group without additional setup".
[7] PKI stands for Public Key Infrastructure

lattice dimension. Furthermore, it has been proven that a randomly chosen instance of a certain lattice problem is at least as hard as the worst-case instance of a related lattice problem, therefore choosing secure keys is easy.

Lattice-based cryptography has been on the cards since the mid-90s. In 2010, Ruckert [46] has proposed the idea of lattice-based blind signatures. Based on the hardness of worst-case lattice problems (such as the Closest Vector Problem and the Smallest Basis Problem), Ruckert’s scheme offers unconditional blindness and one-more-unforgeability in the random oracle model. Let $L$ be the bit-length of the secret key. The scheme remains secure, even if the adversary obtains $L(1-o(1))$ bits of the secret key via arbitrary side channels. This brings the security model closer to reality, where the adversary may obtain information about the secret key, e.g, via (remote) timing attacks or by having physical access to the signing device. According to the author, when applied in e-voting or e-cash schemes, such resilience can help against insider attacks. With its four moves, the scheme is theoretically efficient: all operations have quasi-linear complexity and all keys and signatures require a quasi-linear amount of storage bits.

Ruckert’s work could potentially be a breakthrough in this field. This scheme is expected to withstand even sub-exponential-time and quantum computer attacks, as well as limited side-channel attacks against the secret key, due to the hard lattice problem it is based on. However, it remains to be seen whether this scheme is implementable in practical systems. Secondly, the security has been proved in the random oracle model which is far too idealistic. Further work needs to be put into coming up with a practical version of this scheme, and to see if the security proof can be further strengthened.

## Doing away with the Random Oracle Model

Proving security of blind signature schemes without relying on random oracles is one of the most crucial open problems in research circles today. In the on-going research, a significant step has been taken by Fuchsbauer and Vergnaud in recent months. In [31], they have come up with the first practical fair blind signature scheme whose security poof does not rely on random oracles. It provides a practical electronic voting protocol in the standard model including public verifiability, and compares favorably with other similar systems in terms of computational cost. This scheme can be used to achieve CCA-anonymous group signatures.

The method of security proof adopted in [31] make use of non-interactive zero knowledge proof systems, automorphic signatures and tag-based encryption. If these techniques are explored further, there is a good chance of deriving new ways of building schemes that do not rely on the random oracle model.

## Security of Blind Signatures under Aborts

According to a recent work ([52]), it is important to ‘explore the security of blind signatures under aborts, where the user or the signer may stop the interactive signature issue protocol prematurely’. Most of the previous works on blind signatures discuss security only in regard of completed executions and usually do not impose strong security requirements in case of aborts. To tackle this issue, Camenisch *et al.* ([53]) have introduced the notion of selective failure blindness. To put it in simple terms, a selective-failure blind signature ensures that the blindness property holds even in the case when the signer is able to learn that some executions have aborted. 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.

[52] has derived a technique to turn every secure blind signature scheme into a selective-failure blind signature scheme. This transformation is expected to add only a negligible overhead due to an additional computation of a commitment. It remains to implement this system practically, and to explore whether its security is achievable without the random oracle model.

### Identity-based Blind Signatures

In traditional PKI-based digital signature schemes, certificates generated by a trusted third party are required to ‘bind’ the user’s identity and its public key. In identity-based cryptography, the only secret of each user is its secret identity (e.g. its IP or email address) as a secret key generated by a Key Generation Center (KGC). Hence, in such cryptosystems, the certificates and the intricate management can be avoided.

Due to these advantages, ID-based blind signatures have been an active area of research. In recent past, [50] proposed new partially blind ID based schemes have been proposed whose security has been proven under Diffie Hellman assumption in the random oracle model. Building up on that, the current year has seen the proposal of a highly efficient secure ID-based scheme without Random Oracle in [51]. Research continues to be done in this arena to explore the advantages and short-comings of ID-based schemes to blind signatures.

## 11. Conclusion

In the highly digitized world of today, it is extremely important for every publicly used digital system to provide the right balance between digital anonymity and digital security. A completely secure system may violate the privacy of honest users, while a system that allows complete anonymity to its users may lead to digital crimes that affect both the service-providers and the users. In this regard, blind signature systems have been a pioneering breakthrough in information technology. Most of the currently used e-cash, e-banking and e-voting systems depend heavily on them to keep track of illegal transactions, while making sure that customer privacy is violated to a minimal extent. So far, the ride has been fairly smooth. With changing technologies, new digital attacks emerge but the theory of blind signature schemes keeps adapting simultaneously to combat them. On the other hand, we have observed that the underlying mathematical theory of most of the implementable blind schemes relies on intractable number theoretic problems, which may not be a viable solution in future because computing powers are expected to take a huge leap with time (e.g. in case of advent of quantum computers). This necessitates further research in the domain of blind signatures with increased rigor and vigor, and with special emphasis on developing techniques that are based on *post-quantum* intractable problems.

************************************************

## References

**[1]** D.Chaum, *Blind signatures for untraceable payments*. Advances in Cryptology, Crypto'82, pp.199-203, 1982

**[2]** S. Han and E. Chang, *A pairing-based blind signature scheme with message recovery*. Ardil, C. (ed), Sixth International Enformatika Conference (IEC), pp. 303-308, 2005

**[3]** B. Schoenmakers, *Cryptographic Protocols*. Lecture Notes, Technical University of Eindhoven, 2011

**[4]** D. Chaum, *Blind signature systems.* Advances in Cryptology, Crypto'83, pp.153-156, 1983

**[5]** D. Chaum, *Blinding for unanticipated signatures.* Advances in Cryptology, Eurocrypt'87, pp.227-233, 1987

**[6]** E. Teske, *Full Domain Hash RSA security.* Supplementary Lecture Notes, University of Waterloo, 2011

**[7]** D. Chaum. *Security Without Identification: Transaction Systems to Make Big Brother Obsolete*. Communications of the ACM 28, 10, 1985

**[8]** D. Chaum. *Privacy Protected Payments: Unconditional Payer And/Or Payee Untraceability*. Smartcard 2000, pages 69–93, 1989

**[9]** D. Pointcheval and J. Stern. *Provably secure blind signature schemes*. Advances in Cryptology – ASIACRYPT ' 96, volume 1163, pages 252–265. Springer-Verlag, 1996

**[10]** M. Michels, M. Stadler, and H. Sun. *The security of some variants of the RSA signature scheme*. Computer Security – ESORICS ' 98, volume 1485, pages 85–96. Springer-Verlag, 1998

**[11]** M. Bellare, C. Namprempre, D. Pointcheval, and M. Semanko, *The One-More-RSA-Inversion Problems and* the Security of Chaum's Blind Signature Scheme, Journal of Cryptology, Volume 16(3), pp. 185 - 215, Springer, 2003

**[12]** D. Chaum, A. Fiat and M. Naor, Untraceable Electronic Cash. Advances in Cryptology – Crypto '88. Lecture Notes in Computer Science, Volume 403/1990, pp. 319-327, 1990

**[13]** J. Camenisch, J.-M. Piveteau, and M. Stadler, *Blind signatures based on the discrete logarithm problem*. Advances in Cryptology- Eurocrypt '94, volume 950 of Lecture Notes in Computer Science, pages 428-432. Springer Verlag Berlin, 1994.

**[14]** National Institute of Standards and Technology (NIST). *Digital signature standard (DSS)*. Tech. Rep. FIPS PUB XX, NISS, US Department Commerce, 1993

**[15]** K. Nyberg and R.A Rueppel. *A new signature scheme based on the DSA giving message recovery*. 1st ACM Conference on Computer and Communications Security. pp.58-61, 1993

**[16]** Dominique Schröder, *On the Complexity of Blind Signatures.* PhD Disseration, Technical University of Darmstadt, 2010

**[17]** S. Brands. Untraceable off-line cash in wallet with observers. Advances in Cryptology- CRYPTO '93, LNCS 773, Springer-Verlag, 1993

**[18]** W. Qiu, *How to Construct DLP-Based Blind Signatures and Their Application in E-Cash Systems.* Progress in Cryptography, The Kluwer International Series in Engineering and Computer Science, Volume 769, 73-80, 2004

**[19]** E. Mohammed, A.E. Emarah, and K. El-Shennawy. *A blind signature scheme based on ElGamal signature.* IEEE/AFCEA EUROCOMM 2000 Information Systems for Enhanced Public Safety and Security, pp. 51-53, 2000

**[20]** C.-I Fan and C.-L. Lei. *User efficient blind signatures*. IEEE Electronic Letters, vol. 34, no. 6, pp. 544-546, 1998.

**[21]** Zuhua Shao. *Improved user efficient blind signatures*. Electronic Letters, vol. 36, no. 16, pp. 1372-1374 , 2000.

**[22]** C.-I Fan and C.-L. Lei. *Cryptanalysis on improved user efficient blind signatures*. Electronic Letters, vol. 37, no. 10, pp. 630-631, 2001.

**[23]** J. M. Pollard and C. P. Schnorr. *An efficient solution of the congruence* $x^2 + ky^2 = m \ (mod \ n)$. IEEE Trans. Inf. Theory, vol. 33, no. 5, pp. 702-709, 1987.

**[24]** S.H.V. Solms and D. Naccache, *On blind signatures and perfect crimes*. Computers & Security, pp.581-583, 1992.

**[25]** M. Stadler, J.-M. Piveteau and J. Camenisch, *Fair Blind Signatures*. Advances in Cryptology – Eurocrypt '95. Lecture Notes in Computer Science, Volume 921, 1995

**[26]** S. Micali, *Fair Cryptosystems*. Technical Report MIT/LCS/TR-579.b, 1993.

**[27]** A. Fiat and A. Shamir. *How to prove yourself: Practical Solutions to identification and signature problems*. Proceedings of Crypto '86, LNCS 263, Springer Verlag, pp. 186-194

**[28]** Y. Frankel, Y. Tsiounis, and M. Yung. *"Indirect discourse proofs": Achieving efficient fair off-line e-cash*. Advances in Cryptology — ASIACRYPT '96, volume 1163, Lecture Notes in Computer Science, pages 286–300. Springer-Verlag, 1996.

**[29]** M. Abe and M. Ohkubo, *Provably secure fair blind signatures with tight revocation*. ASIACRYPT 2001, volume 2248 of LNCS, pages 583-602. Springer, 2001.

**[30]** E. Hufschmitt and J. Traore, *Fair blind signatures revisited*. PAIRING 2007, volume 4575 of LNCS, pages 268-292. Springer, 2007

**[31]** G. Fuchsbauer and D. Vergnaud, *Fair Blind Signatures without Random Oracles.* Lecture Notes in Computer Science, Volume 6055, Progress in Cryptology – AFRICACRYPT 2010, Pages 16-33, 2010

**[32]** M. Abe and E. Fujisaki, *How to date blind signatures*. Lecture Notes in Computer Science, Volume 1163, Advances in Cryptology — ASIACRYPT '96, Pages 244-251, 1996

**[33]** M. Abe and T. Okamoto, *Provably Secure Partially Blind Signatures*. Lecture Notes in Computer Science, Volume 1880, Advances in Cryptology — CRYPTO 2000, Pages 271-286, 2000

**[34]** S. M. Chow , C. K. Hui, S. M. Yiu and K. P. Chow, *Two Improved Partially Blind Signature Schemes from Bilinear Pairings.* Lecture Notes in Computer Science, Volume 3574, Information Security and Privacy, pages 355-411, 2005

**[35]** J. Liu, R. Sun and W. Kou, *Fair e-payment protocol based on simple patially blind signature scheme.* Wuhan University Journal of Natural Sciences, Volume 12, Number 1, Pages 181-184, 2007

**[36]** Y. Xie, F. Zhang, X. Chen, and K. Kim, *ID-based Distributed 'Magic Ink' Signature*. Lecture Notes in Computer Science, Volume 2836, pages 249–259. Springer, 2003

**[37]** F. Zhang, R. Safavi-Naini, and C.-Y. Lin, *New Proxy Signature, Proxy Blind Signature and Proxy Ring Signature Schemes from Bilinear Pairings*. Cryptology ePrint Archive, Report 2003/104, 2003

**[38]** T. Zuo-Wen, L. Zhuo-Jun, and T. Chun-Ming, *Digital Proxy Blind Signature Schemes Based on DLP and ECDLP and its Applications*. Technical Report 21, Mathematics-Mechanization Research Center (MMRC), Institute of Systems Sciences, Chinese Academy of Sciences, 2002.

**[39]** S.M. Chow, C.K. Hui, S.M. Yiu, and K.P. Chow, *Forward-Secure Multisignature and Blind Signature Schemes.* Applied Mathematics and Computation, 2004.

**[40]** D. N. Duc, J. H. Cheon, and K. Kim, *A Forward-Secure Blind Signature Scheme Based on the Strong RSA Assumption*. Proceedings, Volume 2836 of Lecture Notes in Computer Science, pages 11–21. Springer, 2003.

**[41]** A. Lysyanskaya and Z. Ramzan, *Group Blind Digital Signatures: A Scalable Solution to Electronic Cash*. Financial Cryptography, Second International Conference, FC 1998, Anguilla, British West Indies, February 23-25, 1998, Proceedings, volume 1465 of Lecture Notes in Computer Science, pages 184–197. Springer, 1998.

**[42]** J. Kim, K. Kim, and C. Lee, *An Efficient and Provably Secure Threshold Blind Signature*. Information Security and Cryptology - ICISC 2001, Fourth International Conference, Seoul, Korea, December 6-7, 2001, Proceedings, volume 2288 of Lecture Notes in Computer Science, pages 318–327. Springer, 2002.

**[43]** D. L. Vo, F. Zhang, and K. Kim, *A New Threshold Blind Signature Scheme from Pairings*. In Symposium on Cryptography and Information Security, SCIS2003, Jan.26-29, 2003, Itaya, Japan, volume 1/2, pages 233–238, 2003.

**[44]** T. K. Chan, K. Fung, J. K. Liu, and V. K. Wei, *Blind Spontaneous Anonymous Group Signatures for Ad Hoc Groups*. Security in Ad-hoc and Sensor Networks, First European Workshop, ESAS 2004.

**[45]** X. Chen, F. Zhang, and K. Kim, *ID-based Multi-Proxy Signature and Blind Multisignature from Bilinear Pairings*. In KIISC conference, Korea, pages 11– 19, 2003.

**[46]** M. Ruckert, *Lattice-Based Blind Signatures*. Lecture Notes in Computer Science, Volume 6477, Advances in Cryptology - Asiacrypt 2010, Pages 413-430

**[47]** C.P. Schnorr, *Efficient Identification and Signatures for Smart Cards*. Crypto '89, LNCS 435, pages 235-251, Springer Verlag, 1990

**[48]** D. Pointcheval, *Strengthened security for blind signatures*. Lecture Notes in Computer Science, Volume 1403, Advances in Cryptology — EUROCRYPT'98, Pages 391-405, 1998

**[49]** D. Galindo1, J. Herranz, and E. Kiltz, *On the Generic Construction of Identity-Based Signatures with Additional Properties*. Asiacrypt’06, LNCS 4284, pp. 178-193. Springer Verlag, 2006

**[50]** W. Chen, B. Qin, Q. Wu, L. Zhang, H. Zhang, *ID-Based Partially Blind Signatures: A Scalable Solution to Multi-Bank E-Cash*. ICSPS ’09 Proceedings of the 2009 International Conference on Signal Processing Systems, IEEE, 2009

**[51]** X. Hu, J. Wang, Y. Yang, *Secure ID-Based Blind Signature Scheme without Random Oracle***.** NCIS '11 Proceedings of the 2011 International Conference on Network Computing and Information Security - Volume 01, IEEE, 2011

**[52]** M. Fischlin and D. Schröder, *Security of Blind Signatures under Aborts*. Lecture Notes in Computer Science, 2009, Volume 5443, Public Key Cryptography – PKC, Pages 297-316, 2009

**[53]** J. Camenisch, G. Neven, and A. Shelat, *Simulatable Adaptive Oblivious Transfer*. Advances in Cryptology, Eurocrypt'07, Lecture Notes in Computer Science, pages 573-590. Springer-Verlag, 2007.