中華民國八十六年全國計算機會議

可應用於電子貨幣上之高效率公平盲目簽章技術
Efficient Fair Blind Signatures for Electronic Cash

范俊逸
Chun-I Fan

台灣大學電機工程系
Department of Electrical Engineering
National Taiwan University
Taipei, Taiwan, R.O.C.
Email: fan@crypto.ee.ntu.edu.tw

雷欽隆
Chin-Laung Lei

台灣大學電機工程系
Department of Electrical Engineering
National Taiwan University
Taipei, Taiwan, R.O.C.
Email: lei@cc.ee.ntu.edu.tw

摘要

在匿名電子貨幣系統中，雖然不可連結性可以保護消費者或使用者的隱私，但是此特性可能被非法者用來進行洗錢及勒索贖金等犯罪行為。公平的盲目簽章技術可以用來解決不可連結性被誤用的問題，在公平盲目簽章系統中，政府或司法機關擁有足夠的資訊可以找出那些犯罪者的身分。本論文提出一個高效率且可應用於電子貨幣上之公平盲目簽章技術。在本系統中，每個簽章只包含兩個整數，而且使用者只需進行數個模運算即可獲得其所要求的簽章。與現有的各系統比較，本系統不僅降低了使用者所需的計算量達 99%，而且也降低了使用者所需的儲存空間達 86%。

關鍵詞：公平盲目簽章，電子貨幣，資訊安全

Abstract

Although the unlinkability property makes it possible to protect the privacy of customers or users in anonymous electronic cash systems, it can be misused by criminals, such as to launder money or to safely get a ransom. The techniques of fair blind signatures are developed to deal with the misuse of unlinkability. The fairness property makes it possible for the government or the judge to find the identities of those criminals. In this paper, we propose an efficient fair blind signature scheme for electronic cash. Only two integers are required to form a signature in our fair blind signature scheme. Furthermore, it only takes several modular computations for a signature requester to obtain and verify a signature. Comparing with the existing schemes proposed in the literatures, our method reduces not only the amount of computations for requesters or users by 99% but also the required storage space for them by 86%.

Keywords: Fair Blind Signatures, Electronic Cash, Information Security.

1. Introduction

Due to the fast progress of network and cryptology technologies, many advanced communication services have been proposed in the literatures to take advantage of the ever-growing networking capabilities. Among these services, electronic cash is a popular one since the technique makes it possible for a payer to pay electronic cash through electronic communication channels. Owing to the ability of protecting the privacy of the payers, blind signature techniques are usually adopted to develop this popular service proposed in the literatures.

A typical blind signature scheme consists of two kinds of participants, a signer and a set of requesters. A requester requests signatures from the signer, and the signer issues blind signatures to the requesters. There are two sets of messages known to the signer: (1) the messages received from requesters for signatures, and (2) the signatures submitted by the requesters for verification later. The key point is that the actual correspondence between these two sets of messages is unknown to the signer. This property is usually referred to as the *unlinkability* property [1, 2, 8, 9, 17]. Due to the unlinkability property, blind signatures have been widely used to construct anonymous electronic cash systems [2, 3].

Since the blind signature techniques provide perfect unlinkability, it is impossible for any one but the requester himself to link a signature to the corresponding instance of the signing protocol which produces that signature. In electronic cash systems, the unlinkability property could be misused by criminals, such as to launder money or to safely get a ransom [23, 24]. To guarantee the quality of this ever-growing popular communication service, modern blind signature techniques should possess the following two properties:

(1). If signature requesters or customers are engaged in legal commercial transactions or payments, their privacy should be protected well; on the other hand, if they misuse the unlinkability property, then the government or the judge should have enough information to uncover their identities. This property is referred to as the *fairness* property.

(2). The addition of the fairness property to blind signatures should not increase the computation load for signature requesters or customers since their computation capacities are limited in some situations such as smart cards and mobile units.

In this paper, we propose an efficient fair blind signature scheme. Our scheme can overcome the misuse of the unlinkability property. With the help of the judge, the government, or a trusted party, it is possible to link a signature to the corresponding instance of the signing protocol. In other words, if the judge or the government does not disclose any necessary information to the signer, the privacy of every requester is protected against the signer. On the other hand, if necessary, the judge or the government can give some appropriate data to the signer or the bank, so that the signer or the bank can link his view of a signature protocol to the signature or cash produced by that protocol, and uncover the identity of the signature requester who requests that signature or cash.

Our scheme only takes several modular computations for a signature requester to obtain and verify a signature, and in the proposed scheme, every signature consists of only 2 integers in Z_n^* where Z_n^* is the set of all positive integers less than and relatively prime to n. Comparing with the existing schemes, our method reduces the amount of computations for signature requesters by 99% and reduces the required storage space for requesters by 86%. Our scheme minimizes both the storage space and the computation load for signature requesters or customers, so that it is suitable for mobile clients and smart-card users.

The rest of the paper is organized as follows. In section 2 and 3, we briefly review the related works in the literatures. In section 4, we present our efficient fair blind signature scheme. The performance of the proposed scheme is examined in section 5. Finally, a concluding remark is given in section 6.

2. Typical blind signatures

The concepts of blind signatures was first introduced by Chaum [2]. Based on the RSA cryptosystem, he proposed a blind signature scheme to achieve the unlinkability property. By means of the techniques of blind signatures, an anonymous electronic cash system was proposed in [3]. In such an electronic cash system, the bank (or the signer) issues electronic cash (e-cash), and a customer (or a requester) can withdraw e-cash from his account, or deposit e-cash into his account in the bank.

Based on the RSA cryptosystem, Ferguson introduced another blind signature scheme tailored for his anonymous electronic cash system proposed in [9]. In [1], the authors proposed a blind signature scheme based on discrete logarithm (DL) problems. In addition to the above scheme, the authors of [1] presented another blind signature scheme based on the Nyberg-Rueppel signature scheme [13]. Based on the Okamoto's protocol of [14] and the Schnorr's protocol of [20], a blind signature scheme was proposed in [17]. The authors of [17] presented another blind signature scheme based on the Okamoto's protocol of [14] and the Guillou-Quisquater protocol of [11].

3. Fair blind signatures

Due to the unlinkability property, the technique of blind signature can protect the privacy of customers in an electronic cash system. Since the technique provides perfect unlinkability, it is impossible for any one but the customer himself to link an e-cash to the corresponding instance of the withdrawing protocol which produces that e-cash. Unfortunately, the unlinkability property could be misused by criminals [23, 24].

Fair blind signatures are developed to cope with the misuse of unlinkability. In an anonymous electronic cash system with fairness property, if customers are engaged in legal commercial transactions or payments, their privacy should be protected well; on the other hand, if they misuse the unlinkability property, then the government or the judge should have enough information to uncover their identities [24].

A fair blind signature scheme consists of three kinds of participants, the judge (or the government), a signer, and a set of requesters. A requester requests signatures from the signer, and the signer issues blind signatures to the requesters. The judge keeps all link information between every instance of the signing protocol and the signature produced by that instance of the protocol. If the judge provides the signer the kept information of some link, then the signer can derive the link. On the other hand, if the judge does not reveal the appropriate link information to the signer, it is computationally infeasible for the signer to derive the link.

In [24], Stadler, Piveteau, and Camenisch proposed three blind signature schemes to achieve the fairness property. The first scheme of [24] is based on the Chaum's blind signature scheme and the cut-and-choose method [2, 3]. The second scheme of [24] is based on a variation of the Fiat-Shamir signature scheme and the concept of one-out-of-two oblivious transfer [7, 10]. The main idea of the third scheme is that the requester has two pseudonyms registered at the judge. One of the pseudonyms is used during the signing protocol, whereas the other one is part of the signature. Thus, the judge, who knows the two corresponding pseudonyms, can link a view of the signing protocol and the corresponding signature.

In the fair blind signature scheme using the cut-and-choose method of [24], a large amount of data is exchanged during the signing protocol, and the resulting signature is large. Although the resulting signature of the fair blind signature scheme using oblivious transfer of [24] is short, it is necessary for a signature requester to perform a large amount of modular computations. Considering the fair blind signature scheme with registration of [24], a large amount of computations is still required for signature requesters.

4. The proposed scheme

In this section, we propose an efficient fair blind signature scheme. Our scheme only takes several modular computations for a signature requester to obtain and verify a signature, and only two integers are required to form a signature in our scheme. Comparing with the existing schemes proposed in the literatures, our method greatly reduces not only the amount of computations for signature requesters but also the required storage space for requesters.

The proposed scheme is based on quadratic residues. Under a modulus n, x is a quadratic residue (QR) in Z_n^* if and only if there exists an integer y in Z_n^* such that $y^2 \equiv_n x$ where Z_n^* is the set of all positive integers which are less than and relatively prime to n. Given x and n, it is computationally infeasible to derive the square root y of x if n contains large prime factors and the factorization of n is unknown [19].

Our proposed fair blind signature scheme consists of four phases: (1) initialization, (2) requesting, (3) signing, and (4) extraction. All the necessary information is published in the initialization phase. To obtain the signature of a plaintext message, a requester requests necessary parameters from the judge, and then submits an encrypted version of the message to the signer in the requesting phase. The signer computes the blind signature of the message and sends it back to the requester in the signing phase. Finally, in the extraction phase, the requester extracts the signature from the blind signature. The details of the proposed fair blind signature scheme are described as follows.

4.1. Initialization

The signer randomly selects two distinct large primes p_1 and p_2 such that $p_1 \equiv_4 p_2 \equiv_4 3$. Then he computes $n = p_1 \cdot p_2$ and publishes n.

The judge randomly chooses two distinct large primes p_3 and p_4 such that $p_3 \equiv_4 p_4 \equiv_4 3$ and $p_3 p_4 > n$, and then computes $\widehat{n} = p_3 \cdot p_4$. The judge publishes $\widehat{n}$ and a string ϖ selected by the judge at random. Let H be a public one-way hashing function [22].

4.2. Requesting

A signature requester randomly chooses three integers y_1, y_2, and y_3 such that for every i with $1 \le i \le 3$,

$$\begin{cases} n < y_i < \widehat{n} < y_i^2, \\ y_i \bmod n \in Z_n^*, \\ y_i \in Z_{\widehat{n}}^*, \text{ and} \\ \varpi \text{ is a prefix of } y_i. \end{cases}$$

Then, the requester computes and submits $(y_i^2 \bmod \widehat{n})$ to the judge for $i = 1$, 2, and 3.

After receiving all $(y_i^2 \bmod \widehat{n})$'s, since the judge has p_3 and p_4, the judge derives the square roots of every $(y_i^2 \bmod \widehat{n})$ in $Z_{\widehat{n}}^*$ [15, 19]. For every i, there exists one square root of $(y_i^2 \bmod \widehat{n})$ in $Z_{\widehat{n}}^*$ with the prefix ϖ, so that the judge can obtain y_1, y_2, and y_3 by finding those square roots with the prefix ϖ.

The judge randomly selects two integers β and γ, and forms $u = H(\beta)$ and $v = H(\gamma)$ such that $((u^2 + v^2) \bmod n)$ is in Z_n^*. Let z be an integer to uniquely identify this instance of the protocol where z is randomly chosen by the judge such that $H(z)$ is a QR in $Z_{\widehat{n}}^*$. The integer z is referred to as the identifier of this instance of the protocol. The judge randomly selects an integer b in Z_n^*, and then computes

$$\begin{cases} \widehat{b} = y_1^{-1} \cdot b \bmod n \\ \widehat{u} = y_2^{-1} \cdot u \bmod n \\ \widehat{v} = y_3^{-1} \cdot v \bmod n. \end{cases}$$

The judge derives a square root $\widehat{z}$ of $H(z)$ in $Z_{\widehat{n}}^*$ such that $(\widehat{z})^2 \equiv_{\widehat{n}} H(z)$, and then the judge sends the tuple $\left(\widehat{b}, \widehat{u}, \widehat{v}, \widehat{z}, z\right)$ to the requester. In addition, the judge stores the tuple (β, γ, b, z) in its database.

After receiving $\left(\widehat{b}, \widehat{u}, \widehat{v}, \widehat{z}, z\right)$, the requester can obtain b, u, and v by computing

$$\begin{cases} (y_1 \cdot \widehat{b} \bmod n) = b \\ (y_2 \cdot \widehat{u} \bmod n) = u \\ (y_3 \cdot \widehat{v} \bmod n) = v. \end{cases}$$

To request a signature of a plaintext m, the requester computes $\alpha = (H(m) \cdot (u^2 + v^2) \bmod n)$. The requester submits the tuple $(\alpha, z, \widehat{z})$ to the signer.

Table 1: Property Comparisons.

	Our Scheme	[1][1]	[2]	[9]	[17][1]	[24][2]
Foundation	QR	DL/DL	RSA	RSA	RSA/DL	RSA/DL/DL
Randomization	Yes	Yes/Yes	No	Yes	Yes/Yes	No/Yes/Yes
Unlinkability	Yes	Yes/Yes	Yes	Yes	Yes/Yes	Yes/Yes/Yes
Fairness	Yes	No/No	No	No	No/No	Yes/Yes/Yes

[1] Two schemes in [1] and [17]. [2] Three schemes in [24].

The signer examines whether $(\widehat{z})^2 \equiv_{\widehat{n}} H(z)$ or not. If true, the signer randomly selects an integer δ and computes $x = H(\delta)$ such that $(\alpha \cdot (x^2+1) \bmod n)$ is a QR in Z_n^*. Then, the signer sends $(x, z, \widehat{z})$ to the judge.

After verifying that $(\widehat{z})^2 \equiv_{\widehat{n}} H(z)$, the judge retrieves the stored tuple (β, γ, b, z) through the identifier z, and computes $c = ((ux+v)(u-vx)^{-1} \bmod n)$. where $u = H(\beta)$ and $v = H(\gamma)$. The judge checks if the integer c is different from all the other c's which are recorded by the judge during all previous instances of the protocols. If not, the judge requests the signer to choose another integer x until $c = ((ux+v)(u-vx)^{-1} \bmod n)$ is unique among all the recorded integers c's. If yes, the judge computes $\lambda = (b^2 \cdot (u-vx) \bmod n)$, and sends λ to the signer. Then, the judge records the tuple (β, γ, b, z, c).

4.3. Signing

After receiving λ, the signer computes $e = (\lambda^{-1} \bmod n)$ and derives an integer t in Z_n^* [15, 19] such that

$$t^4 \equiv_n \alpha \cdot (x^2+1) \cdot e^2.$$

Then, the signer sends the tuple (e, t, x) to the requester, and stores (δ, z, id) in his database where id is the identity of the requester.

4.4. Extraction

After receiving the tuple (e, t, x), the requester computes

$$\begin{cases} s = b \cdot t \mod n \\ c = b^2 \cdot e \cdot (ux+v) \mod n. \end{cases}$$

Thus, (s, c) is a signature of m. To verify the signature (s, c) of m, one can examine if

$$s^4 \equiv_n H(m) \cdot (c^2+1).$$

4.5. Discussions

In the requesting stage of the scheme, the signer receives two integers α and λ from the requester and the judge for requesting a signature of a plaintext m, where

$$\begin{cases} \alpha = H(m) \cdot (u^2+v^2) \bmod n \\ \lambda = b^2 \cdot (u-vx) \bmod n \end{cases}.$$

Then in the extraction stage of the scheme, the requester obtains a signature (s, c) of m by computing

$$\begin{cases} s = b \cdot t \bmod n \\ c = b^2 \cdot e \cdot (ux+v) \bmod n \end{cases}$$

where $t^4 \equiv_n \alpha \cdot (x^2+1) \cdot e^2$ and $e \equiv_n b^{-2} \cdot (u-vx)^{-1}$. The signer cannot link the tuple (α, λ) to the signature (s, c) of m because the integers (u, v, b) are randomly selected and kept secret by the judge and the requester in the scheme.

Consider the linkage recovery in our scheme. Given a signature $(\widetilde{s}, \widetilde{c})$ of a plaintext $\widetilde{m}$ produced by some instance of the protocol, the judge can retrieve the unique tuple (β, γ, b, z, c) with $c = \widetilde{c}$ from its database. Hence, the signature $(\widetilde{s}, \widetilde{c})$ of $\widetilde{m}$ is produced by the instance of the protocol with identifier z. If the judge reveals the tuple (β, γ, z, c) to the signer, then the signer can retrieve the tuple (δ, z, id) through the identifier z from his database. Thus, $c \equiv_n (H(\beta)H(\delta) + H(\gamma))(H(\beta) - H(\gamma)H(\delta))^{-1}$. Hence, the signer can obtain not only the instance z of the protocol which produces $(\widetilde{s}, \widetilde{c})$ but also the identity id of the requester who requests that signature of $\widetilde{m}$ in that instance of the protocol. Therefore, if the judge reveals appropriate information to the signer, the link between an instance of the signing protocol and the corresponding signature produced can be established by the signer. On the other hand, given the signature $(\widetilde{s}, \widetilde{c})$ of the plaintext $\widetilde{m}$, the signer cannot find the tuple (δ, z, id) from his database without the identifier z given by the judge.

In our scheme, the signer perturbs the message received from every requester before he signs it by using a random integer x. This is usually referred to as the *randomization* property [9]. A randomized blind signature scheme can withstand the chosen-text attacks [21]. Our scheme and the blind signature schemes of [1, 9, 17] possess the randomization property, while the blind signature scheme of [2] does not have this property. In addition, given an integer

Table 2: Performance Comparisons with Fair Blind Signature Schemes.

	Our Scheme[1]	[24][2]	[24][3]	[24][4]
No. of Exponentiation Computations	0	40	240	10
No. of Inverse Computations	0	1	0	1
No. of Hashing Computations	2	60	2	2
No. of Multiplications	18	60	160	6
Computations Reduced :		99.85%	99.97%	99.45%
No. of Messages Transmitted	14	72	81	12
No. of Integers in a Signature	2	40	2	6
Space Reduced :		86%	81%	11%

[1] An on-line judge is needed in our scheme.

[2] The first scheme of [24]. [3] The second scheme of [24]. [4] The third scheme of [24].

c and a plaintext m, let s be an integer such that $s^4 \equiv_n H(m) \cdot (c^2+1)$. Thus, s is a 4th root of the integer $(H(m) \cdot (c^2+1) \bmod n)$ in Z_n^*. Since n contains large prime factors, computing a 4th root of an integer in Z_n^* is computationally infeasible without the factorization of n [19].

The comparisons of properties between our scheme and the existing schemes of [1, 2, 9, 17, 24] are summarized in table 1.

5. Performance

Typically, under a modulus n, the computation time for a modular exponentiation operation is about $O(|n|)$ times that of a modular multiplication where $|n|$ denotes the bit length of n [22]. The modulus n is usually taken from 512 bits to 1024 bits in a practical implementation [22]. In [4, 6, 12], some fast exponentiation algorithms are proposed. In [6], it requires $0.3381|n|$ modular multiplications and large amount of storage, e.g. 83370 stored values for a 512-bit modulus, to compute a modular exponentiation computation. An enhanced version of [6] is introduced in [4]. However, it still requires $0.3246|n|$ modular multiplications and large amount of storage, e.g. 36027 stored values for a 512-bit modulus, to compute a modular exponentiation computation [4]. The algorithm of [12] needs $(1.164|e| + 3)$ modular multiplications to compute $(x^e \bmod n)$ where $|e|$ is the bit length of e and $|e|$ has to be large enough (say 128 bits) in the RSA-type blind signature schemes to resist possible low-exponent attacks [5, 25].

In our fair blind signature scheme, no exponentiation and inverse computations are performed by signature requesters. Moreover, only several modular additions and multiplications are required for a requester to obtain and verify a signature.

In the fair blind signature schemes of [24], many modular exponentiation computations and inverse computations are needed for the requesters to obtain and verify signatures, while these time-consuming computations are not required in our scheme. Comparing with the fair blind signature schemes of [24], our scheme reduces the amount of modular computations for signature requesters by 99% under a 1024-bit modulus. The comparisons of the numbers of modular computations performed by a requester between our scheme and the fair blind signature schemes of [24] are summarized in table 2. The comparisons of the storage space required for requesters between our scheme and the schemes of [24] are also summarized in table 2.

Furthermore, compared to the unfair blind signature schemes of [1, 2, 9, 17], our method still largely reduces the amount of the modular computations for signature requesters.

6. Conclusion

In this paper, we have proposed an efficient fair blind signature scheme. Our scheme not only possesses the fairness property, but also minimizes the computation load and the storage space required for signature requesters. Hence, the proposed scheme is suitable for the situations where hardwares and computation capacities of signature requesters or customers are limited. With the help of the judge or the government, it is possible to link a signature to the corresponding instance of the signing protocol, so that the identities of signature requesters or customers who misuse the unlinkability property can be uncovered in our scheme.

References

[1] J. L. Camenisch, J. M. Pivereau, and M. A. Stadler, "Blind signatures based on the discret logarithm problem," *Advances in Cryptology-*

EUROCRYPT'94, LNCS 950, Springer-Verlag, 1995, pp. 428-432.

[2] D. Chaum, "Blind signatures systems," *Advances in Cryptology-CRYPTO'83*, Plenum, 1983, p. 153.

[3] D. Chaum, A. Fiat, and M. Naor, "Untraceable electronic cash," *Advances in Cryptology-CRYPTO'88*, LNCS 403, Springer-Verlag, pp. 319-327, 1990.

[4] C. Y. Chen, C. C. Chang, and W. P. Yang, "Hybrid method for modular exponentiation with precomputation," *Electronics Letters*, vol. 32, no. 6, 1996, pp. 540-541.

[5] D. Coppersmith, M. Franklin, J. Patarin, and M. Reiter, "Low-exponent RSA with related messages," *Advances in Cryptology-EUROCRYPT'96*, LNCS 1070, Springer-Verlag, 1996, pp. 1-9.

[6] V. Dimitrov and T. Cooklev, "Two algorithms for modular exponentiation using nonstandard arithmetics," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, Vol. E78-A, no. 1, 1995, pp. 82-87.

[7] S. Even, O. Goldreich, and A. Lempel, "A randomized protocol for signing contracts," *Communications of the ACM*, vol. 28, 1985, pp. 637-647.

[8] C. I. Fan and C. L. Lei, "A multi-recastable ticket scheme for electronic elections," *Advances in Cryptology-AISACRYPT'96*, LNCS 1163, Springer-Verlag, 1996, pp. 116-124.

[9] N. Ferguson, "Single term off-line coins," *Advances in Cryptology-EUROCRYPT'93*, LNCS 765, Springer-Verlag, 1994, pp. 318-328.

[10] A. Fiat and A. Shamir, "How to prove yourself: Practical solutions to identification and signature problems," *Advances in Cryptology-CRYPTO'86*, LNCS 263, Springer-Verlag, 1986, pp. 186-194.

[11] L. C. Guillou and J. J. Quisquater, "A practical zero-knowledge protocol fitted to security microprocessor minimizing both transmission and memory," *Advances in Cryptology-EUROCRYPT'88*, LNCS 330, Springer-Verlag, 1988, pp. 123-128.

[12] D. C. Lou and C. C. Chang, "Fast exponentiation method obtained by folding the exponent in half," *Electronics Letters*, vol. 32, no. 11, 1996, pp. 984-985.

[13] K. Nyberg and R. A. Rueppel, "A new signature scheme based on the DSA giving message recovery Schemes," *The first ACM Conference on Computer and Communications Security*, November 3-5, Fairfax, Virginia.

[14] T. Okamoto, "Provably secure and practical identification schemes and corresponding signature schemes," *Advances in Cryptology-CRYPTO'92*, Springer-Verlag, LNCS 740, 1992, pp. 31-53.

[15] R. C. Peralta, "A simple and fast probabilistic algorithm for computing square roots modulo a prime number," *IEEE Transactions on Information Theory*, vol. 32, no. 6, 1986, pp. 846-847.

[16] S. Pohlig and M. E. Hellman, "An improved algorithm for computing logarithms over GF(p) and its cryptographic significance," *IEEE Transactions on Information Theory*, vol. 24, 1978, pp. 106-110.

[17] D. Pointcheval and J. Stern, "Provably secure blind signature schemes," *Advances in Cryptology-ASIACRYPT'96*, LNCS 1163, Springer-Verlag, 1996, pp. 252-265.

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

[19] M. O. Rabin, "Digitalized signatures and public-key functions as intractable as factorization," *Technical Report*, MIT/LCS/TR212, MIT Lab., Computer Science, Cambridge, Mass. Jan. 1979.

[20] C. P. Schnorr, "Efficient identification and signatures for smard cards," *Advances in Cryptology-CRYPTO'89*, Springer-Verlag, LNCS 435, 1990, pp. 235-251.

[21] A. Shamir and C. P. Schnorr, "Cryptanalysis of certain variants of Rabin's signature scheme," *Information Processing Letters*, vol. 19, 1984, pp. 113-115.

[22] G. J. Simmons, *Contemporary Cryptology: The Science of Information Integrity*, IEEE Press, N.Y., 1992.

[23] S. V. Solms and D. Naccache, "On blind signatures and perfect crime," *Computer and Security*, vol. 11, pp. 581-583, 1992.

[24] M. Stadler, J. M. Piveteau, and J. Camenisch, "Fair blind Signatures," *Advances in Cryptology-EUROCRYPT'95*, LNCS 921, Springer-Verlag, pp. 209-219, 1995.

[25] M. J. Wiener, "Cryptanalysis of short RSA secret exponents," *IEEE Transactions on Information Theory*, vol. 36, 1990, pp. 553-558.