An Exploitation of Blind Signature Schemes to Simulate Privacy-related Applications

Kazi Md. Rokibul Alam, Saifuddin Mahmud and Mohammad Nazmul Alam Khan
Department of Computer Science and Engineering,
Khulna University of Engineering and Technology
Khulna-9203, Bangladesh

ABSTRACT

This paper presents the simulation of privacy-related applications employing blind signature (BS) schemes. Two popular privacy-related applications: 'traditional BS based electronic voting system (EVS)' and 'traditional offline electronic payment protocol (EPP)' have been chosen here. A BS scheme is a cryptographic protocol that plays a vital role to conduct the electronic transactions of privacy-related applications securely but anonymously. It ensures the confidentiality of the private information of a user while she involves in an electronic transaction over the internet. Intuitively, existing BS schemes can be categorized as traceable and untraceable. RSA cryptosystem based two popular schemes from two categories: the scheme of Chaum [1] from traceable schemes and the scheme of Hwang et al. [2] from untraceable schemes have been chosen here for simulation. The upshot of the simulation model is the comparison of computation time requirement of blinding, singing, unblinding and verification operations involved in different steps of privacy-related applications evaluated by the chosen BS schemes.

General Terms

Cryptography, RSA Cryptosystem.

Keywords

Blind Signature Scheme, Untraceability, Electronic Voting System, Electronic Payment Protocol.

1. INTRODUCTION

A Blind signature (BS) scheme is exploited to provide anonymity in privacy-related applications like electronic voting system (EVS), electronic payment system etc. In electronic transactions a BS scheme is typically employed because herein the signer and the message owner (i.e. the user) are different parties. Unlike a normal digital signature scheme, in a BS scheme, the signer signs on the blinded message using his private key where the message is blinded by the user. Therefore the signer knows nothing about the content of the message [2]. Later on, anyone can publicly verify the legitimacy of the signature using the signer's public verification key. Here, the signer of the BS scheme is unable to link the message-signature pair even when the signature has been revealed to the public [6]. Thus a BS scheme ensures the authenticity of a message.

In privacy-related applications, usually the user demands a BS from the signer. For this, first she blinds her message. Then the signer sings on the blinded message from which later on, the user can generate the signed message. A BS scheme assures that a user is unable to create the signed messages by itself. Also the execution of a BS scheme can create at most one unblinded signed message [2]. Moreover the signed message generated by a BS scheme ensures that neither it can be forged nor it can be traced. For this reason, when a BS scheme is exploited, the authenticity of the signed message can be verified but the origin of the signed message cannot be traced.

An ideal BS scheme is supposed to satisfy the following requirements [2, 4]:

1. ***Correctness***: the correctness of the signature of a message signed by a BS scheme can be checked by anyone using the signer's public key.
2. ***Blindness***: the content of the message should be blind to the signer; the signer of the BS scheme is unable to see the content of the message.
3. ***Unforgeability***: the signature is the proof of the signer, and no one else can derive any forged signature and pass verification.
4. ***Untraceability***: the signer of the BS scheme is unable to link the message-signature pair even when the signature has been revealed to the public

Already mentioned that, existing BS schemes can be categorizes as traceable (e.g. [1]) and untraceable (e.g. [2]). It is found that Chaum's scheme [1] can achieve only the first three requirements; whereas Hwang et al.'s scheme [2] is capable to satisfy all the above requirements including untraceability. The comment presented in [8] also has proved that Hwang et al.'s scheme is untraceable. Although traceable and untraceable both schemes can be applied in privacy-related applications, untraceable scheme is definitely admired because the signer is absolutely unable to link the message-signature pair. As a result it is more secure. In this paper, a simulation of 'traditional BS based EVS' and 'traditional offline EPP' employing both traceable and untraceable BS schemes have been performed. The outcome of the simulation model is the evaluation of computation time requirement of blinding, singing, unblinding and verification phases of the chosen BS schemes.

The outline of this paper is as follows: Section II describes the chosen privacy-related applications i.e. a traditional BS based EVS presented in [10] and a traditional offline EPP presented in [3]. Section III explains the BS schemes chosen for simulation i.e. the scheme proposed by Chaum [1] and the scheme proposed by Hwang et al. [2]. The results of simulation have been discussed in Section IV. Finally, concluding remarks are explained in Section V.

2. PRIVACY-RELATED APPLICATIONS

2.1 Blind Signature based Electronic Voting System

EVS (in this scope, the user, the signer and the message are denoted as the voter, the election authority and the electronic vote, respectively), is a process in which voters cast their votes while a group of election authority collects the votes and outputs the final tally. In a traditional BS based EVS, a transaction between the voter and the election authority starts by creating an electronic vote by the voter. Then the voter blinds its vote and sends it to the authority requesting a signature on it. After verifying the validity of voter, the authority digitally signs on the blinded vote, and sends it back to the voter. Finally after unblinding the signed vote, voter verifies the legitimacy of the signature on its vote.

A traditional BS based EVS as explained in [10] is discussed below.

Step 1: Voter creates its vote and picks a secret random number to blind the vote.
Step 2: Voter proves its authenticity to the election authority and sends its blinded vote for requesting the signature on it.
Step 3: Election authority verifies the validity of the voter, then signs on the blinded vote and sends it back to the voter.
Step 4: Now voter unblinds the signed vote and verifies the legitimacy of the signature on the unblinded signed vote.
Step 5: Finally voter broadcasts this signed vote through an anonymous channel [10] during the voting stage to compute the election result.

2.2 Offline Electronic Payment Protocol

An EPP (in this scope, the user, the signer and the message are denoted as the customer, the Bank and the electronic coin, respectively), is a protocol that allows the exchange of electronic coin and goods between the customer and the merchant with value assured by the Bank's signature where the identity of the customer is kept concealed [7]. In an offline EPP, electronic payment between the customer and the merchant is performed in offline state; therefore no online link to the Bank is required at every time, and it enables to detect over spending [5]. Here, the customer and the merchant should have individual accounts in the Bank, [5].

A traditional offline EPP as explained in [3] is discussed below. It consists of three major phases i.e. withdrawal, payment and deposit phases. The procedure assumes that the customer wants to purchase some goods from the merchant and that they have accounts with the Bank.

2.2.1 Withdrawal

1. The customer creates an electronic coin and blinds it.
2. The customer sends the blinded coin to the Bank with a withdrawal request.
3. Bank digitally signs on the blinded coin.
4. Bank sends the signed-blinded coin to the customer and debits her account.
5. The customer unblinds the signed coin.

2.2.2 Payment

1. The customer gives the merchant the coin.
2. The merchant verifies the Bank's digital signature. (optional)
3. The merchant gives the customer the merchandise.

2.2.3 Deposit

1. The merchant sends the coin to the Bank.
2. Bank verifies Bank's digital signature.
3. Bank verifies that the coin has not already been spent.
4. Bank enters the coin in spent-coin database.
5. Bank credits the merchant's account.

3. BLIND SIGNATURE SCHEMES

3.1 Chaum's Scheme

Chaum's BS scheme [1] is based on RSA cryptosystem and consists of five phases, these are: initializing, blinding, signing, unblinding, and verifying. The BS scheme (in the following, the voter/customer, the election authority/the Bank and the electronic vote/coin are denoted as the user, the signer and the message, respectively) is described briefly as follows.

3.1.1 Initialization

The signer randomly chooses two large primes p and q, and computes $n = p * q$ and $\varphi(n) = (p - 1) * (q - 1)$. Then he chooses two large numbers e and d such that $ed \equiv 1 \bmod \varphi(n)$ and $GCD(e, \varphi(n)) = 1$. Let (e, n) be the signer's public key and d be the signer's private key for signing. The signer keeps (p, q, d) secure and publishes (e, n).

3.1.2 Blinding

The user has a message m, and she wishes to have it signed by the signer. The user randomly selects an integer r as the blinding factor. The user computes and submits the integer $\alpha = r^e * m \bmod n$ to the authority.

3.1.3 Signing

After receiving α from the user, the signer computes and sends the integer $t = \alpha^d \bmod n$ to the user.

3.1.4 Unblinding

After receiving t from the signer, the user computes $s = t * r^{-1} \bmod n$.

3.1.5 Verifying

As a result, s is the signature on the message m. Now anyone can verify the legitimacy of the signature by checking whether $s^e \equiv m \bmod n$.

It is easy to see that Chaum's BS scheme cannot meet the requirement of untraceability. Here the above description suggests that, the signer should keep a pair of records (α_i, t_i) for every blinded message. When the user reveals k pairs of (m_i, s_i) to the public, the signer can compute $k * r'_i = t_i * s_i^{-1} \bmod n$ according to each stored pair of i values (α_i, t_i), where $i = 1, 2, 3, ..., k$. Then the signer can trace the BS by checking whether each r'_{i} and $r'_{(i-1)}$ have the same relation. It is assumed that each user has its own random generator to generate an integer r by a relation in Chaum's BS scheme. In [2] it has been shown that Chaum's BS has this weakness. Thereby Chaum's BS scheme cannot achieve perfect untraceability.

3.2 Hwang et al.'s Scheme

To overcome the weakness of untraceability of Chaum's BS scheme, a new untraceable scheme was proposed in [2]. It is also based on RSA cryptosystem and also consists of five phases: initializing, blinding, signing, unblinding, and verifying. The signer first publishes the public information in the initializing phase. In the blinding phase, the user blinds the message and sends it to the signer for requesting the signature. Then the signer signs on the blinded message in the signing phase. In the unblinding phase, the user derives the signature from the blinded signature. Finally, anyone can verify the legitimacy of the signature in the verifying phase. The scheme is described as follows.

3.2.1 Initialization

The phase is the same as the initializing phase in Chaum's BS scheme. The signer keeps (p, q, d) secure where d is the signer's key for signing and publishes (e, n) as public key.

3.2.2 Blinding

The user has a message m, and she wishes to have it signed by the signer. For this purpose, the user randomly selects two distinct integers' r_1 and r_2 as the blinding factors. Then she randomly chooses two primes a_1 and a_2 such that the greatest common divisor (GCD) of $a1$ and $a2$, denoted by GCD (a_1, a_2), is 1. Then, the user computes the blinded messages $\alpha_1 = r_1^e * m^{a1} \bmod n$ and $\alpha_2 = r_2^e * m^{a2} \bmod n$, and sends ($\alpha_1$, α_2) to the signer.

3.2.3 Signing

After receiving (α_1, α_2) from the user, the signer randomly chooses two primes b_1 and b_2 such that the GCD of b_1 and b_2, denoted by GCD (b_1, b_2), is 1 and signs the blinded message by computing $t_1 = \alpha_1^{(b1d)} \bmod n$ and $t_2 = \alpha_2^{(b2d)} \bmod n$. Then the signer sends them back to the user along with (b_1, b_2). Note that (t_1, t_2, b_1, b_2) denote the blinded signatures.

3.2.4 Unblinding

After receiving (t_1, t_2, b_1, b_2) from the signer, the user computes a_1b_1 and a_2b_2. Due to the four primes (a_1, a_2, b_1, b_2) where GCD (a_1, a_2) = 1 and GCD (b_1, b_2) = 1, GCD (a_1b_1, a_2b_2) is also equal to 1. When GCD (a_1b_1, a_2b_2) = 1, there must be exactly two integers w and t that satisfy the equation $a_1b_1w + a_2b_2t = 1$. It is called the Extended Euclidean algorithm. The four parameters (a_1, a_2, w, t) are kept secret by the user. Then the user computes $s_1 = t_1 * r_1^{-b1} = m^{a1b1d} \bmod n$ and $s_2 = t_2 * r_2^{-b2} = m^{a2b2d} \bmod n$. Then the user can derive the signature s by computing $s = s_1w * s_2t \bmod n$ and then publishes (m, s).

3.2.5 Verifying

As a result, s is the signature on the message m. Now anyone can verify the legitimacy of the signature by checking whether $s^e \equiv m \bmod n$.

3.3 Discussions

3.3.1 Blindness

Blindness is the main property for a BS scheme. Blindness means that the signer can sign on the message without knowing what the value of the message is containing. In Chaum's BS scheme [1], the user picks a blinding factor r to compute the blinded message $\alpha = r^e * m \bmod n$. Hence, the signer does not get to know the message m. Similarly, in Hwang et al.'s BS scheme [2], the user picks four blinding factors (r_1, r_2, a_1, a_2) to compute the blinded message $\alpha_1 = r_1^e * m^{a1} \bmod n$ and $\alpha_2 = r_2^e * m^{a2} \bmod n$. Therefore, the signer still cannot know the message m.

3.3.2 Untraceability

Untraceability is another important property for a BS scheme. For any given valid signature (m_i, s_i), the signer is unable to link this signature to the message. In Hwang et al.'s scheme [2], the signer can be kept from tracing the BS. The demonstration is as follows. The signer keeps a set of records (α_{1i}, α_{2i}, t_{1i}, t_{2i}, b_{1i}, b_{2i}) for every blinded message. However, when the user reveals (m_i, s_i) to the public, the signer has no way to get any information (r'_{1i} and r'_{2i}) from these records. He cannot trace the relation between r_{1i} and r_{2i}. In addition, s consists of s_1 and s_2, neither of which the signer knows. Furthermore, without the knowledge of the secure integers (a_{1i}, a_{2i}, w_i, t_i, r_{1i}, r_{2i}), the signer cannot trace the BS [2].

4. EXPERIMENTAL ANALYSIS

4.1 Experimental Setup

A simulation of traditional BS based EVS and traditional offline EPP considering Chaum's traceable scheme [1] and Hwang et al.'s untraceable scheme [2] has been developed, and the computation times required for blinding, signing, unblinding and verification operations have been measured. The environment consists of a 2.53 GHz CPU with 04 GBytes of RAM, and Microsoft Visual Studio 2010 [9] with .Net Framework 4, 1024 bit modulus running on Windows7 operating system is used for encryption/signature. In the table all computation times do not consider the communication time. The different operations of privacy-related applications that are not related to cryptography have not been considered.

4.2 Experimental Results

Table 1. Computation time requirement for Chaum's scheme

Operations	Time (ms)
User blinding its message	09
Signer signing the blinded message	32
User unblinding the signed message	08
Anyone verifying the signature	34

Table 1 shows the computation time requirement of blinding, signing, unblinding and verification operations employing Chaum's traceable scheme. In traditional EVS/EPP, the major cryptographic operations are to blind a message by the user, to sign on the blinded message by the signer, to unblind the signed message by the user and the verification of the BS by any third party. Here to blind a message requires 09ms where the message blinding by using unknown random number which is known as the blinding factor is prepared in advance, therefore its computation time is not considered. Then to sign on the blinded message by the signer, it requires 32ms. Later on to unblind the signed message by the user, it takes 08ms. Lastly the verification of the BS by any third party using the signer's public verification key requires 34ms.

Table 2. Computation time requirement for Hwang et al's scheme

Operations	Time (ms)
User blinding its message	347
Signer signing the blinded message	77
User unblinding the signed message	309
Anyone verifying the signature	34

Table 2 shows the computation time requirement of blinding, signing, unblinding and verification operations employing Hwang et al.'s untraceable scheme. In traditional EVS/EPP, the cryptographic operations are as same as Chaum's scheme as mentioned in the above paragraph. However the manners of cryptographic techniques to conduct the operations are different. Here to blind a message by the user, it requires 347ms. At this point for blinding a message, two unknown random numbers which are also known as the blinding factors and two random prime numbers are required and it is assumed that they are prepared in advance. Therefore their computation time is not considered. Then to sign on the blinded message by the signer, it requires 77ms. Here, the signer randomly chooses two prime numbers according to the technique described in [2] and it is assumed that they are prepared in advance. Therefore their computation time is not considered also. Later on to unblind the signed message by the user, it takes 309ms. Finally the verification of the BS by any third party using the signer's public verification key requires 34ms.

Table 3. Comparison between Chaum's and Hwang et al.'s schemes

Operations	Chaum's Scheme	Hwang et al.'s scheme
	Time (ms)	
Blinding a message	09	347
Signing a message	32	77
Unblinding a signed message	08	309
Verifying the signature	34	34

Table 3 shows the comparison of computation time requirement between Chaum's scheme and Hwang et al.'s scheme. Here to blind a message, Chaum's scheme takes only 09ms whereas Hwang et al.'s scheme takes 347ms. To use as a blinding factor, Chaum's scheme generates only one random integer whereas Hwang et al.'s scheme generates two distinct random integers. Moreover it considers another two prime numbers in its blinding phase. In the case of signing a blinded message, Chaum's scheme takes only 32ms whereas Hwang et al.'s scheme takes 77ms. Here, Chaum's scheme signs only by using the secret key whereas Hwang et al.'s scheme uses another two primes with the secret signing key. To unblind a signed message, Chaum's scheme takes only 08ms whereas Hwang et al.'s scheme takes 309ms. In this case, Hwang et al's scheme also needs to consider lots of parameters than Chaum's scheme. Finally to verify the signature, Chaum's scheme takes 34ms and Hwang et al.'s scheme takes 34ms. In this point, the comparison of computation shows that although the time requirement of various operations by Hwang et al.'s untraceable scheme is greater than Chaum's traceable scheme, Hwang et al.'s scheme is preferable than Chaum's scheme with respect to security.

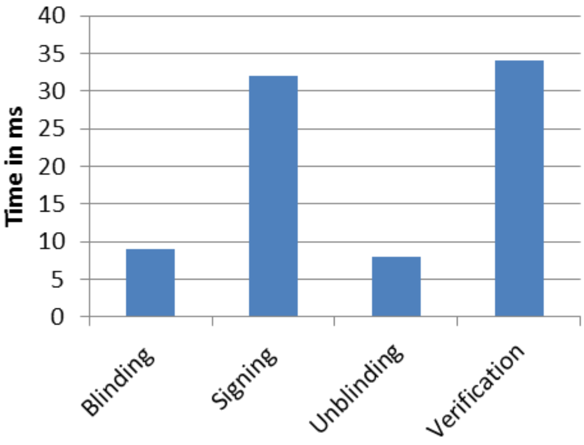

Fig 1: Computation time required by Chaum's scheme

Fig 1 shows the computation time requirement of handling various operations of BS of a traditional EVS/EPP employing Chaum's scheme. The Fig. shows that to blind a vote, it requires 09ms. Then, to sign on a blinded message takes 32ms. Next to unblind the signed message takes 08ms. Lastly, the verification of the signature requires 34ms.

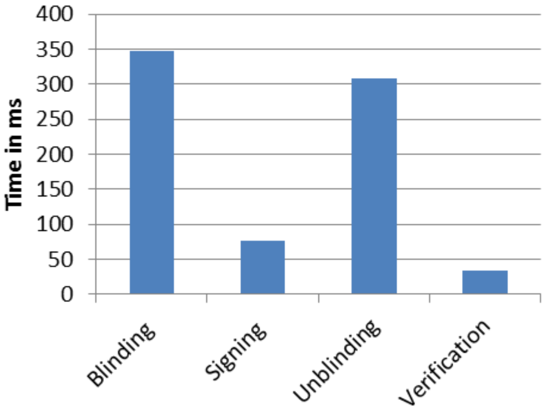

Fig 2: Computation time required by Hwang et al.'s scheme

Fig 2 shows the computation time requirement to handle various operations of BS of a traditional EVS/EPP employing Hwang et al.'s scheme. The Fig. shows that to blind a message it takes 347ms. Then to sign on a blinded message, it requires 77ms. Next to unblind the signed message takes 309ms. Finally the verification of the signature requires 34ms.

5. CONCLUSIONS

This paper has performed the simulation of two privacy-related applications i.e. 'a traditional BS based EVS' and 'a traditional offline EPP' employing Chuaum's BS scheme [1] and Hwang et al.'s BS scheme [2] that evaluates the computation time requirements of cryptographic operations involved in these applications. Although untraceability is an essential but tough requirement for a BS scheme, the literature review (e.g. [2, 8]) shows that Hwang et al.'s scheme is untraceable whereas Chaum's scheme is traceable. Here the comparison of computation time requirement shows that Hwang et al.'s scheme requires much time that Chaum's scheme to conduct the simulation, however Hwang et al.'s scheme is certainly admired. The reason is, it fully satisfies all the requirements of an ideal BS scheme. Therefore it is efficient to exploit Hwang et al.'s scheme in privacy related applications of cryptography.

6. REFERENCES

[1] D. Chaum, "Blind signatures system," Advances in Cryptology, CRYPTO'83, pp. 153−156, 1983.

[2] M. Hwang, C. Lee, and Y. Lai, "An untraceable blind signature scheme," in IEICE Trans. Fundamentals, Vol. E86−A, No. 7, pp. 1902−1906, 2003.

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

[4] D. Jena, S. K. Jena, and B. Majhi, "A novel untraceable blind signature based on elliptic curve discrete logarithm problem," in International Journal of Computer Science and Network Security, Vol. 7, No. 6, pp. 269–275, 2007.

[5] Sattar J Aboud, "Secure e–payment protocol," in International journal of Security (IJS), Vol. 3, No. 5, pp. 85–92, 2009.

[6] M. Kwon and Y. Cho, "Randomization enhanced blind signature schemes based on RSA," in IEICE Trans. Fundamentals, Vol. E82, No. 1, pp. 1−4, 1999.

[7] H. Oros and C. Popescu “A Secure and Efficient Off-line Electronic Payment System for Wireless Networks”. International Journal of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844 Vol. V, No. 4, pp. 551-557, 2010.

[8] Fang-Ping Chiang Yi-Mien Lin and Ya-Fen Chang “Comments on the Security Flaw of Hwang et al.’s Blind Signature Scheme” in International Journal of Network Security, Vol.6, No.2, pp.185–189, Mar. 2008

[9] Microsoft Visual Studio available at http://www.microsoft.com/visualstudio/en-us/products/2010-editions

[10] K. Sampigethaya and R. Poovendran, “A Framework and Taxonomy for Comparison of Electronic Voting Schemes,” Elsevier Computers and Security, Vol. 25, pp. 137-153, 2006.