How to Date Blind Signatures

Masayuki Abe
abe@isl.ntt.jp

Eiichiro Fujisaki
fujisaki@sucaba.isl.ntt.jp

Tel: +81 468 59 2570 Fax: +81 468 59 3858
NTT Information and Communication Systems Laboratories
Nippon Telegraph and Telephone Corporation
1-2356 Take, Yokosuka-shi, Kanagawa, 238-03 Japan

Abstract. A blind signature provides perfect confidentiality to a message and signature pair. Due to this feature, the blind signature has one downside; the signer can not assure himself that the blinded message accurately contains the information he desires. In a practical sense, it is essential for the signer to include some term of validity in the signing message to prevent abusing. Of course the term must not violate the confidentiality of the message. This paper discusses partial blinding of a signed message. We consider RSA and it is proved that forging the proposed scheme by multiple signing is as difficult as breaking RSA. The strategy can be also applied to those blind signature schemes that use a trapdoor function. An electronic cash system is shown as an application of the proposed scheme. Unlike most privacy-protected electronic cash system, it successfully minimizes the growth of the bank's database.

1 Introduction

The blind signature, first introduced by D.Chaum in [1], is a scheme that yields a signature and message pair whose information does not leak to the signer. The scheme was implemented using RSA. Several blind signature schemes [2] have been developed since then and it has been proven that blind signatures could be realized by using any commutative random self-reducible problem [3].

Due to its confidentiality, blind signatures have been mainly used for electronic cash protocols [4,5,6,7,8] as a tool to protect customer's privacy.

In these protocols, the *personal information* which distinguishes messages or message holders must be kept from the signer's view. However, the signer must assure himself that the message contains accurate information without seeing it. The well known Cut and Choose mythology [4] introduced by D.Chaum can solve this problem. However, the vast amount of data needed to obtain sufficient security spoils its efficiency. S.Brands proposed a restrictive blind signature in [6] as a part of an electronic cash protocol. In his scheme, the signer signs a message which contains message holder's public key as a message holder can blind it only to a restricted form. A verifier checks the signer's signature, and then requests the message holder to endorse the transaction by signing with the secret key behind the public key in the message. If the message does not contain the correct public key, the endorsement fails. This scheme is believed to provide one solution for including personal information in an application where the customer must prove possession of the authorized secret to the verifier without showing the secret.

Another kind of check is the information that must be clearly indicated to the signer to check its validity. It is the *common information* among the players, of which the proof of possession is insufficient. One example is the term of signature validity. If a signature is not dated, it lasts as long as the signer's key lasts. This might be a problem in some applications. So, in practice, a way of dating the blind signature is needed. The other example, from an electronic cash scheme, is the amount of the payment (or a coin). It must be clearly checked between a bank, a customer, and a shop. Since the perfect confidentiality makes it difficult to include common information in the message, it is applied to the signer's key so that a key represents a piece of common information. Unfortunately, this approach narrows the variation of the common information because key management becomes too complex.

This paper proposes a partially blinded signature scheme where a part of the message is kept in the clear while the rest is kept in secret. Section 2 discusses the basic idea. An RSA-based scheme is developed in section 3. Security of the scheme and the possibility of applying it to other signature schemes are also discussed in section 3. In section 4, an electronic cash system that minimizes the size of the bank's database is shown as an application of the scheme.

2 Basic Idea

In order to make our goal clear, the blind signature scheme should be described.

Let x and $f(x)$ be the signer's private key and public key, respectively.$S(x,m)$is the signer's signature to message m with private key x. The set $\{f(x),m,S(x,m)\}$ satisfies the verification equation $V(f(x),m,S(x,m))$.

Let $B(m,r)$ be a blinded message which is statistically or perfectly indistinguishable from m as long as blinding factor r is not revealed. Similarly, let $U(S,r')$ be an unblinded signature which is statistically or perfectly indistinguishable from S as long as blinding factor r' is not revealed.

The blind signature protocol is described as below.

(1) The sender sends a blinded message $B(m,r)$ to the signer.

(2) The signer signs the message with his secret key, and then sends the signature $S(x,B(m,r))$ to the sender.

(3) The sender checks if the signature satisfies the verifying function. Then unblinds the signature by calculating $U(S(x,B(m,r)),r')$ which is equal to $S(x,m)$. She then sends the unblinded signature and the message to the verifier.

(4) The verifier checks that the unblinded signature pair satisfies the verifying function.

Our goal is to add a constant c as common information to both the blinded signature triple and the unblinded one. Accordingly, the partially blinded signature set is $\{f(x), c, B(m,r), S(x,c,B(m,r))\}$ and the unblinded signature set is $\{f(x), c, m, S(x,c,m)\}$.

Moreover, there are two conditions:

(1) no one can forge c.

(2) no information except c can be transmitted from the signer to the verifier.

To follow the first condition, c must not work with $B(m,r)$ in the signing function. If not, the effect of c to the signature can be cancelled by unblinding. Thus the constant must work outside the scope of blinding or unblinding. However, if there exists an inverse signing function involving c, it is still easy for the sender to cancel c's effect and pretend c' instead.

Using a trapdoor function can solve this problem easily. Accordingly, instead of directly using c in the signing function, the signer uses the hidden constant $\bar{c}$ produced by the inverse function of the trapdoor. Since only the signer can calculate $\bar{c}$, it is hard for others to cancel $\bar{c}$'s effect from the signature. Using the trapdoor function, c works as if a part of the signer's public key. The signer opens the trapdoor backwards, and computes the corresponding secret key $x = f^{-1}(c)$.

A protocol based on RSA is developed in the next section.

3 Protocol

3.1 Description

Let c be common information whose length is $k-2$ bits. The function $\tau(c)$ calculates

$$\tau(c) = 2^{k-1} + 2h(c) + 1. \qquad (1)$$

where $h(\cdot)$ is an one-way function. $\tau(c)$ is a formatting function designed to keep its domain in $2^{k-1} < \tau(c) < 2^k$ so that $\tau(c_i)$ does not divide $\tau(c_j)$ where $i \neq j$. Also, it is designed to produce odd numbers only so that it becomes relatively prime with λ. This prevents a kind of forgery by getting multiple signatures described latter.

N is a product of two large primes p and q. N satisfies

$$s_i / \lambda \text{ for all prime } s_i \ (3 \le s_i \le 2^k - 1),$$

where λ is the LCM of $p-1$ and $q-1$. The prime e is an RSA public exponent which is larger than or equal to $2^k - 1$. The corresponding private key is d given by $ed = 1 \bmod \lambda$.

The partial blind signature protocol is as follows.

(1) The sender and the signer negotiate and agree on the constant c. Or it could be a common constant like the current date so that they can produce c independently.

(2) The sender randomly chooses a blind factor $R \in Z_N^*$ and blinds a message M by $Z = MR^{e\tau(c)}$, then sends Z to the signer. At this point, $e\tau(c)$ is a public key that contains common information.

(3) The signer calculates corresponding private key d_c by $d_c = 1/e\tau(c) \bmod \lambda$. His blinded signature is $\Phi = Z^{d_c} \bmod N$, and he sends it to the sender.

(4) Receiving the signature Φ, the sender recovers the signature to the bare message M by $S = \Phi / R \equiv M^{d_c} \bmod N$.

In the next subsection, we discuss the security of this protocol.

3.2 Security Considerations

Two types of cheating against the partial blind signature is considered here.

(1) The sender changes the negotiated constant c.

(2) The signer includes a hidden message to distinguish the transaction later.

First, type (1) is discussed. If an exponent $\tau(c_i)$ can be expressed by a polynomial P of any possible $\tau(c_j)$ as in

$$\tau(c_i) = P_i(\tau(c_j)), \tag{2}$$

the sender can forge the constant c. For example, let $\tau(c) = P_i(\tau(c-1)) = \tau(c-1) + 2$ as well as $\tau(\cdot)$ as is true in the previous subsection. Then the correct signature S to the message M with the constant c works as if it is also the correct signature to the message MS^2 with the constant $c-1$.

To prevent this kind of forgery, it is necessary to construct a signing message M from the intended message m through a one-way function $h(\cdot)$. The verifier then requests message m' that satisfies $MS^2 = h(m')$ that the sender can not show. As a result, the strength of the protocol against the forgery of this type depends on the strength of the specified one-way function.

Still, there is the possibility of forgery type (1) by getting multiple signatures to the same message M with different constants. Suppose the sender finds a combination of c_j which satisfies

$$S_i^{e\tau(c_i)} \equiv S_i^{\Pi e\tau(c_j)} \equiv M \bmod N \tag{3}$$

for some c_i.

The sender then gets the signature S_j to the message M with c_j. He then requests the signer to sign the previous signature S_j as the message with $c_{j'}$. By repeating this protocol for all j, the sender can obtain the signature S_j that passes verification for M with c_i.

However, once the forgery is successful, it means that the sender can cheat any constants. See the following lemma.

Lemma 1: If the sender can find a combination of c_j which satisfies equation (3) for some c_i, M and given N, he can then compute signatures for any constant c_k.

Proof 1: If the sender can find a combination of c_j which satisfies equation (3), the sender knows $\tau(c_i)$ and $\tau(c_j)$ that satisfies

$$e\tau(c_i) \equiv \prod e\tau(c_j) \bmod \lambda. \tag{4}$$

Then there exists a constant w that follows

$$e\tau(c_i) \equiv \prod e\tau(c_j) + w\lambda. \tag{5}$$

Since $\tau(\cdot)$ satisfies

$$\tau(c_i) \nmid \tau(c_j) \text{ for all } i \text{ and } j \text{ where } i \neq j, \tag{6}$$

w can not be zero in equation (5).

Therefore, the sender obtains $w\lambda$ by subtracting $e\tau(c_i)$ from $\prod e\tau(c_j)$. Then by calculating $d{=}1/e\tau(c_k) \bmod w\lambda$, the sender can compute any equivalent secret exponent for any c_k.

(Q.E.D.)

The original RSA signature is to compute a signature S for some message M with a fixed secret exponent d and N. As described in lemma 1, the successful forger can compute any secret exponent including the one used for original RSA signature. So we conclude that this forgery is at least as hard as breaking the original RSA signature.

Considering the case if $\tau(c_i) = \Pi(c_j)/\Pi(c_k)$ holds, the one-way function prevents a forger to determine particular combination of c_j and c_k. Therefore as well as the former case, the strength against the forgery of this type depends on the strength of the specified one-way function. When implementing, the length of $\tau(c_i)$ should be kept long to some sort so that the possibility of collision becomes small.

There should be a discussion of the type (2) threat. Because constant c is clear to the sender and is negotiated at the beginning of or before the protocol, the signer can not include any information in c. Note that the sender must not agree to include any one-time elements in c which can distinguish the transaction.

Since the blindness holds to the message and signature sets as well as original RSA blind signature does, the signer and the verifier can not distinguish the blinded and unblinded signature sets.

The above forgery may not be inclusive for the common information. A similar discussion can be seen in [11,12]. Especially, [12] discusses some generalized RSA signature schemes which also apply the message to the exponent.

3.3 Application to other signature schemes

The reason why the RSA-based protocol uses the signer's public exponent is to ensure interoperability with the current public key certificate format based on X.509[13]. Therefore, for all theoretical intents, it is possible to omit the signer's public exponent e for simplicity. That is, the secret key is derived only from the agreed information by opening the trapdoor backward.

Once the agreed information takes over the portion of the signer's public key, it is similar to an ID-based public key system [9,10] where the key distribution center derives the signer's secret key from his well known identity while the proposed scheme derives it from the agreed information. So any ID based public key system which can construct a blind signature protocol can be used for yielding a partial blind signature protocol.

4 Application

In this section, one application of the partial blind signature to an efficient electronic cash system is discussed. Several privacy-protected electronic cash systems have been introduced, but the size of the database needed to store all the used coins must be basically infinite in order to detect double spending. This discourages bankers from putting this kind of system into practice. Against this problem, the proposed scheme is dramatically efficient as well as retaining its privacy protecting feature.

Figure 1 briefly depicts the withdrawal protocol of the limited database system. The bank's role is to issue a kind of ticket, which is exchangeable with corresponding electronic coins, receive the ticket and issue electronic coins. The other role of the bank, which is not depicted in Figure 1, is to verify paid coins.

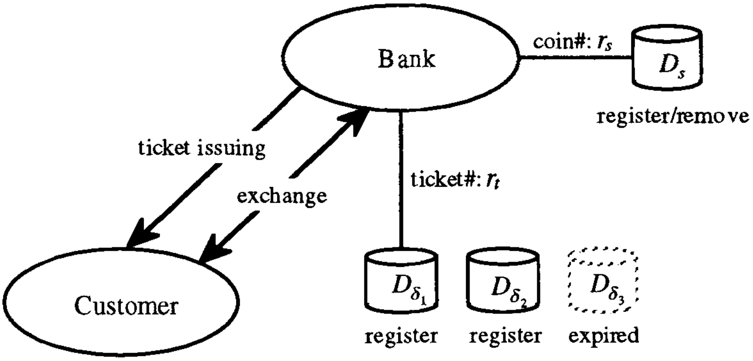

Fig. 1. Concept of Limited Database Electronic Cash System

The scenario is described below. Before the protocol starts, the bank distributes secure hash function $h(\cdot)$ and its public key modulus N used for tickets. For simplicity, the bank's public exponent is omitted from the protocol in subsection 3.1. Correspondingly, the public key $\{e_0, N_0\}$ used for coins is distributed.

(1) Customer sends a blinded message Z as

$$Z = h(r_t)R^{\tau(\alpha\|\delta_1)} \bmod N$$

where r_t is a unique random number, α is the amount to withdraw and δ_1 is today's date.

(2) The bank calculates corresponding secret key $d_{\alpha\delta_1}$ as

$$d_{\alpha\delta_1} = 1/\tau(\alpha\|\delta_1) \bmod \lambda(N).$$

Then signs Z and returns

$$\Phi = Z^{d_{\alpha\delta_1}} R \bmod N.$$

(3) The customer recovers signature S onto $h(r_t)$.

Within the specified period, the ticket S must be exchanged for electronic coins. The exchange procedure is as follows.

(4) The customer sends the ticket S and necessary data $\{\alpha, \delta_1, r_t\}$ to the bank.

(5) The bank verifies S with the corresponding verification function and check its validity with δ_1. If δ_1 has already expired, the ticket can not be accepted. Or, the bank searches database D_{δ_1} in order to make sure that r_t has not already been exchanged. If not, register r_t, issues a coin

$$C = h(\alpha, r_s, h(r_t))^{d_0} \bmod N_0,$$

and registers unique number r_s to database D_s.

When the customer uses the coin, the protocol runs as below.

(6) The customer sends C, α, r_s and r_t to a merchant.

(7) The merchant checks if they are correct as a coin, and passes them to the bank.

(8) The bank searches database D_s. If r_s is found, the bank returns OK and removes r_s from D_s. Otherwise returns NG to the merchant.

Thus the database D_s does not hold used coin entry. Moreover, after pre-determined relatively short period, for example a week, the unexchanged tickets issued on δ_i expire. The bank can then delete entire data in D_{δ_i}. These removal of the used coins and expired tickets limits the maximum size of the bank's database.

A on-line electronic cash system was discussed here, however, it is possible to make it an off-line type by replacing $h(r_t)$ with the customer's registered public key. In that case, the customer's privacy can be protected by pseudo individuality, i.e. his anonymous public key.

5 Conclusion

The concept of a new type of blind signature scheme has been introduced. The scheme allows the signer to add some information which the sender has approved beforehand into blinded signatures. It prevents the transfer of latent messages from the signer to the verifier without the sender. This new feature provides a way to stop the sender abusing the signature as well as assuring the sender that there is no hidden information which may infringe his privacy.

The scheme was successfully implemented on RSA. It has been proved that forging a signature by multiple signature attack is as hard as breaking RSA signature scheme. The possibility of applying the same strategy to ID based signature schemes has been shown.

An electronic cash system was described that uses the proposed scheme. The system prevents the unlimited growth of the bank's database, a well-known problem of previous electronic cash systems.

It remains as further work to construct a protocol that uses non-trapdoor signature schemes.

References

[1] D.Chaum: Blind Signatures for Untraceable Payments, *Advances in Cryptology -Proceedings of Crypto'82*, Plenum Press, 1983, pp. 199-203.

[2] D.Chaum, T.Pedersen: Wallet Databases with Observers, *Advances in Cryptology -CRYPTO'92*, LNCS 740, Springer Verlag, pp. 89-105.

[3] T.Okamoto, K.Ohta: Divertible zero-knowledge interactive proofs and commutative random self-reducibility, *Advances in Cryptology - EUROCRYPT '89*, LNCS 434, Springer- Verlag, pp. 134-149.

[4] D.Chaum, A.Fiat, M.Naor: Untraceable Electronic Cash, *Advances in Cryptology - CRYPTO '88*, LNCS 403, Springer Verlag, pp. 319-327.

[5] D.Chaum: Online Cash Checks, *Advances in Cryptology - EUROCRYPT'89*, LNCS 434, Springer-Verlag, pp. 288-293.

[6] S.Brands: Untraceable Off-line Cash in Wallets with Observers, *Advances in Cryptology - CRYPTO'93*, LNCS 773, Springer -Verlag, pp. 302-318.

[7] T.Okamoto, K.Ohta: Universal Electronic Cash, *Advances in Cryptology - CRYPTO '91*, LNCS 576, Springer-Verlag, pp. 324-337.

[8] T.Okamoto: An Efficient Divisible Electronic Cash Scheme, *Advances in Cryptology - CRYPTO'95*, LNCS 963, Springer, pp. 438-451.

[9] A.Shamir: Identity-Based Cryptosystems and Signature Schemes, *Advances in Cryptology - Proceedings of CRYPTO'84*, Springer-Verlag, pp.47-53.

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

[11] Ganesan R., Y.Yacobi: A Secure Joint Signature and Key Exchange System, *Bellcore Technical Memorandum*, TM-ARH-1994.

[12] W.Jonge, D.Chaum: Some Variations on RSA Signatures & Their Security, *Advances in Cryptology - Proceedings of CRYPTO'86*, Springer-Verlag, pp.49-59.

[13] CCITT Recommendation X.509: The Directory-Authentication Framework, Consultation Committee, International Telephone and Telegraph, International Telecommunications Union, Geneva, 1989.