1-out-of-n Oblivious Signatures

[†]Raylin Tso, Takeshi Okamoto, and Eiji Okamoto

[†]Department of Computer Science,
National Chengchi University,
No.64, Sec.2, ZhiNan Rd., Taipei City, 11605, Taiwan.

Department of Risk Engineering
Graduate School of Systems and Information Engineering
University of Tsukuba,
1-1-1 Tennodai, Tsukuba, Ibaraki, 305-8573, Japan,

[†]raylin@cs.nccu.edu.tw
ken@risk.tsukuba.ac.jp
okamoto@risk.tsukuba.ac.jp

Abstract. An oblivious signature with n keys (or messages) is a signature that the recipient can choose one of n keys (or messages) to get signed while the signer cannot find out on which key (or message) the recipient has got the signature. This kind of signature is firstly introduced by L. Chen in 1994. However, the previous reference does not crisply formalize the notion. Furthermore, the proposed constructions are less efficient in both communication and computation. In this paper, we first give formal definitions on the model of oblivious signatures. Then, based on the Schnorr signature, we propose our efficient oblivious signature scheme. A comparison result is also provided in this paper which shows that our scheme is more efficient than Chen's schemes and those using a combination of a signature scheme and an oblivious transfer protocol.

Key words: 1-out-of-n signature, oblivious signature, oblivious transfer, Schnorr signature.

1 Introduction

Nowadays, as consumers increasingly rely on the internet for shopping, banking and other activities, privacy has become more and more important for consumers who worry about how personal information is used. Motivated by the increasing interest in issues relating to the protection of personal privacy, L. Chen in 1994 proposed the concept of oblivious signatures [4].

In [4], L. Chen considered two types of oblivious signature schemes. The first one is an oblivious signature scheme with n keys and the second one is an oblivious signature scheme with n messages.

Oblivious signature with n keys could be considered a complement of group signature [5]. This scheme is a multiparty protocol. The participants are a group of n signers $S_1, \cdots, S_n$ (or a signer with n different keys) and a recipient R. The scheme has the following characteristics.

1. By executing the protocol, the recipient R can get a message signed with one of n keys which is chosen by himself.
2. The possible signers, even the holder of the accepted key, cannot find out with which key the message is got signed.
3. When necessary, R can show that he has got a signature (on the message) with one of the n keys without revealing with which special one.

The application of this scheme is made in the case of accessing sensitive databases. In this case, a database can only be accessed with a permit (which is possibly a signature signed by the administrator of the database) from the administrator of the database. But the information about which database interests the user may be sensitive. So the user chooses n databases and get the permit (ie., a signature) for only one of n databases without revealing which one.

The second scheme involves a signer S and a recipient R. The common input for both R and S contains n messages. The scheme has the following characteristics.

1. By executing the protocol, the recipient R can choose only one the the n messages to get signed.
2. The signer cannot find out on which message the recipient has got the signature.
3. When necessary, R can show that he has got a signature on one of the n messages without revealing which special one.

This scheme is useful in the case of internet shopping. In this case, the user will buy a software from the seller. In order to prevent illegitimate use, the software can be used if and only if it is signed by the seller. However, the information about which product interests the user may be sensitive in some stage. So the user can choose n softwares and get one and only one signed by the seller without revealing which one.

Motivation Although the concept of oblivious signatures is not completely new, previous references do not crisply formalize the notion, the model of the schemes as well as the security model. Moreover, previously proposed constructions are less efficient in both communication and computation. For example, in L. Chen's schemes [4], to generate an oblivious

signature, a round of interaction consisting of three movies between a signer and a recipient is required. Overall overhead during the interaction of the schemes is at least $3072n$ bits, where n is the number of keys or messages. The size of the generated signature is also a problem. In L. Chen's schemes, a signature consists of 9 parameters with more than 7000 bits in total.

Our Contributions Motivated by the above mentioned problems, in this paper, we first give formal definitions on the model of oblivious signature schemes and give the security requirements of the scheme. Then, based on the Schnorr signature [14], we propose our efficient oblivious signature scheme. A comparison result is also provided in this paper which shows that our scheme is more efficient than Chen's schemes and those using a combination of a signature scheme and an oblivious transfer protocol.

1.1 Related Works

Oblivious Transfer Oblivious transfer was firstly introduced in 1981 by M. Q. Rabin [13]. It is a protocol by which a sender sends some information to the receiver while remains oblivious to what is received. Theoretically, an oblivious signature can be implemented by an oblivious transfer. However, to construct an oblivious signature in this way is inefficient.

Blind Signature In cryptography, a blind signature, as introduced by D. Chaum [3], is a form of digital signature which allows a user to get a message signed by a signer without revealing *any information* about the message to the signer. The resulting blind signature can be publicly verified in the manner of a regular digital signature. Examples of applications include digital election systems and digital cash schemes.

Note that we cannot use a blind signature to construct an oblivious signature on n messages since, in this case, there is no guarantee that the message getting signed is actually the one of n predetermined messages. In other words, a recipient may construct some other messages on which the signer is not going to offer the signature. Furthermore, an oblivious signature with n keys cannot be generated from a blind signature since there is only one signer with one private/public key-pair in a blind signature schemes.

2 Preliminaries

This section gives some cryptographic primitives and definitions required for our construction.

Definition 1. Discrete Logarithm (DL) Problem: Let G be a cyclic group of prime order p and g be a generator of G. The DL problem to the base g means the following problem:

$$\text{Given} \quad g, h \in G, \ \text{where } h = g^x \bmod p \text{ for some unknown } x, \quad \text{find } x.$$

The DL problem is believed to be difficult and also to be the hard direction of a one-way function.

Definition 2. Forking Lemma[12]: Let $(\mathcal{K}, \mathcal{S}, \mathcal{V})$ be a digital signature scheme with security parameter 1^k, with a signature of the form $(m, \sigma_1, h, \sigma_2)$, where $h = \mathcal{H}(m, \sigma_1)$ and σ_2 depends on σ_1 and h only. Let $\mathcal{A}$ be a probabilistic polynomial time Turing machine whose input only consists of public data and which can ask $q_h > 0$ queries to the random oracle. Assume that, within time bound T, $\mathcal{A}$ produces, with probability $\epsilon \geq 7q_h/2^k$, a valid signature $(m, \sigma_1, h, \sigma_2)$. Then, a replay of the attacker $\mathcal{A}$, where interactions with the signer are simulated, outputs two valid signature $(m, \sigma_1, h, \sigma_2)$, and $(m, \sigma_1, h', \sigma_2')$ such that $h \neq h'$, within time $T' \leq 84480Tq_h/\epsilon$.

3 Definition of Oblivious Signature

In this section, we formally define the notion of an oblivious signature and the security requirements behind the oblivious signature. One important modification to the original definitions in [4] is that the characteristic of *ambiguous verification*[1] is neglected in our definition. This is because that, by implementing the technique of *universal* 1-*out-of-n signatures* recently proposed by R. Tso et. al. in [15], this characteristic can be easily achieved for *any* three-move type signature schemes [7]. Consequently, we should not consider ambiguous verification as a unique feature of obvious signatures. It should be considered as an additional feature for all three-move type signatures.

There are two types of oblivious signatures: a 1-out-of-n oblivious signature with n keys and a 1-out-of-n oblivious signature with n messages.

[1] The characteristic that, when necessary, the recipient can show that he has got a signature with one of n keys (or messages) without revealing with which special one.

Due to the space limitations, we only give the formal model for 1-out-of-n oblivious signatures with n messages. However, we emphasize that the formal model for 1-out-of-n oblivious signatures with n keys can be easily defined accordingly.

3.1 Formal Model of Oblivious Signatures with n Messages

A 1-out-of-n oblivious signature scheme (abbreviated to $\mathcal{OS}_1^n$) with n messages involves three types of entities: a recipient $\mathcal{R}$, an oblivious signer $\mathcal{S}$ and a verifier $\mathcal{V}$.

- **A recipient $\mathcal{R}$:** for any input of n messages, $m_1, \cdots, m_n$, $\mathcal{R}$ can choose any one of these n messages to get signed by $\mathcal{S}$.
- **An oblivious signer $\mathcal{S}$:** $\mathcal{S}$ is able to sign the message chosen by $\mathcal{R}$ but is not able to learn which one of the n messages is actually signed.
- **A verifier $\mathcal{V}$:** $\mathcal{R}$ converts the oblivious signature into a generic signature σ and transmits σ to $\mathcal{V}$. $\mathcal{V}$ is able to verify the validity of the signature without any secrete information.

The following definition gives the formal model of a $\mathcal{OS}_1^n$.

Definition 3. An oblivious signature protocol $\mathcal{OS}_1^n$ consists of four tuples, $(\mathcal{G}, \mathcal{S}, \mathcal{R}, \mathcal{V})$, where $\mathcal{S}, \mathcal{R}$ are two interactive Turing Machines, $\mathcal{G}$ is a probabilistic polynomial-time algorithm and $\mathcal{V}$ is a deterministic polynomial-time algorithm.

- $(para, pk, sk) \leftarrow \mathcal{G}(1^k)$**:** A probabilistic polynomial-time algorithm $\mathcal{G}$ which takes a security parameter 1^k as input and the output is the public parameters, $para$. Based on $para$, a public/private key pair (pk, sk) of a signer $\mathcal{S}$ can be defined accordingly.
- $(completed/notcompleted, \sigma/fail) \leftarrow interact(\mathcal{S}(para, pk, sk, m_1, \cdots, m_n), \mathcal{R}(para, pk, m_l))$**:** $\mathcal{S}$ and $\mathcal{R}$ are a pair of polynomially-bounded probabilistic interactive Turing machines. Both machines have the following tapes: a read-only input tape, a write-only output tape, a read/write work tape, a read-only random tape, a read-only communication tape and a write-only communication tape. The input tape of $\mathcal{S}$ is given the public parameter, $para$, the public/private key pair (pk, sk) and n messages $(m_1, \cdots, m_n)$. The input tape of $\mathcal{R}$ is given $para, pk$ and a message $m_l \in \{m_1, \cdots, m_n\}$. The length of all inputs must be polynomial in the security parameter 1^k of the algorithm $\mathcal{G}$. Then $\mathcal{S}$ and $\mathcal{R}$ engage in the interactive protocol of some polynomial number of rounds. At the end of this protocol, the output tape of $\mathcal{S}$

contains either *completed* or *notcompleted* and the output tape of $\mathcal{R}$ contains either fail (if $\mathcal{S}$ outputs "*notcompleted*") or a signature σ (if $\mathcal{S}$ outputs "completed") where σ is a generic signature on the message m_l.

- $1/0 \leftarrow \mathcal{V}(\sigma, para, pk, m_l)$**:** A deterministic polynomial-time algorithm $\mathcal{V}$ which takes the signature σ, the public parameters $para$, a public key pk of a signer $\mathcal{S}$ and a messages m_l as input, returns 1 or 0 for accept or reject, respectively.

3.2 Security Requirements

We now define the securities required for oblivious signatures. The securities defined in this section are modified from the security definitions of blind signatures defined in [1] and [9].

In the coming definitions, $negl(n)$ denotes any function which grows slower than $\frac{1}{n^c}$ for sufficiently large n and some constant c.

Definition 4. (Completeness) If $\mathcal{S}$ and $\mathcal{R}$ follow the signature issuing protocol properly and, at the end of the protocol, $\mathcal{S}$ outputs completed and $\mathcal{R}$ outputs σ, then, with probability at least $1 - negl(n)$, σ satisfies $\mathcal{V}(\sigma, para, pk, m_l) = 1$. The probability is taken over the coin flips of $\mathcal{G}, \mathcal{S}$ and $\mathcal{R}$.

The signature σ on messages m_l is said valid with regard to $(para, pk)$ if it leads $\mathcal{V}$ to accept.

Except the completeness of the protocol, in order to define security, we discuss separately protecting the recipient and the signer. The security for signers is the unforgeability of signatures and the security for recipients is the ambiguity in selected messages (against signers). The security for signers is protected in the sense of computational security and the security for recipients is protected in the sense of unconditional security.

To define the security for signers, we first introduce the following game.

Definition 5. (Game A) Let $\mathcal{R}^*$ be a probabilistic polynomial time forging algorithm. $\mathcal{R}^*$ executes the recipient's part and tries to forge a new signature σ^* on a message, m^*.

1. $(para, pk, sk) \leftarrow \mathcal{G}(1^k)$,
2. $\mathcal{R}^*(para, pk)$ engages in the signature issuing protocol with $\mathcal{S}(para, pk, sk)$ for any message-set $\mathcal{M}_i$ and any message $m_{(i,j)} \in \mathcal{M}_i$ which are adaptively chosen by $\mathcal{R}$. This step can be executed in polynomially

many number of times where $\mathcal{R}^*$ can decide in an adaptive fashion when to stop. In each execution, when $\mathcal{S}$ outputs completed, then $\mathcal{R}^*$ obtains a valid signature σ_i on the message $m_{(i,j)} \in \mathcal{M}_i$. Let t denote the number of executions where $\mathcal{S}$ outputs completed, and $m_{(i,j_i)}$ denote the message corresponding to the signature σ_i, $1 \leq i \leq t$.

3. $\mathcal{R}^*$ outputs a new signature σ^* on a message, m^*, where $m^* \notin \{m_{(1,j_1)}, \cdots, m_{(t,j_t)}\}$.

Definition 6. (Security for signers:Unforgeability) An oblivious signature scheme provides the security for signers if, for any probabilistic polynomial-time forging algorithm $\mathcal{R}^*$ that plays the above game, we have

$$Pr\left(\mathcal{V}(\sigma^*, para, pk, m^*) = 1\right) < negl(n).$$

The probability is taken over the coin flips of $\mathcal{G}, \mathcal{S}$ and $\mathcal{R}^*$.

Intuitively, the security for signers means that, except the signatures σ_i on $m_{(i,j_i)}$, $1 \leq i \leq t$, it is *computationally* infeasible, without the knowledge of the private key sk of $\mathcal{S}$, to produce any valid signature which will be accepted by $\mathcal{V}$. This part is similar to the notion of Existential Unforgeability against Adaptive Chosen Message Attack (EUF-ACMA) [8] for any standard publicly verifiable signature scheme.

The security for recipients is defined through Game B.

Definition 7. (Game B) Let $\mathcal{S}^*$ be an attacking algorithm with unlimited computation power which executes the signer's part and $\mathcal{R}$ be an honest recipient that follows the signature issuing protocol. Let m_0, m_1 be two messages randomly picked by $\mathcal{S}^*$ and $\mathcal{M} \supseteq \{m_0, m_1\}$ be a set of messages which is also determined by $\mathcal{S}^*$. Let $b \in_R \{0, 1\}$ which is kept secret from $\mathcal{S}^*$. The message m_b is put on the input tape of $\mathcal{R}$ which is also kept secret from $\mathcal{S}^*$. The purpose of $\mathcal{S}^*$ is to predict b via the execution of the following game.

1. $(para, pk, sk) \leftarrow \mathcal{G}(1^k)$,
2. $\{m_0, m_1\} \leftarrow \mathcal{S}^*(para, pk, sk, \mathcal{M})$,
3. $\mathcal{S}^*$ engages in the signature issuing protocol with $\mathcal{R}(para, pk, m_b), b \in_R \{0, 1\}$,
4. $\mathcal{S}^*$ outputs a bit $b' \in \{0, 1\}$ according to the view from steps 1, 2, and 3 (ie., $\mathcal{S}^*$ is not allowed to view the output of $\mathcal{R}$ at the end of the signature issuing protocol).

We say that the attacking algorithm $\mathcal{S}^*$ wins the game if $b' = b$.

Definition 8. (Security for recipients against signers:Ambiguity in selected messages) An oblivious signature scheme provides unconditional security for recipients against signers if, for any attacking algorithm $\mathcal{S}^*$ executing the signer's part, $\mathcal{S}^*$ wins in Game B with probability at most $1/2 + negl(n)$. The probability is taken over the coin flips of $\mathcal{G}$, $\mathcal{R}$ and $\mathcal{S}^*$.

Intuitively, the security for recipients against signers means that it is *unconditionally* infeasible for any attacker $\mathcal{S}^*$ to find out which one of the messages in $\mathcal{M}$ is chosen by a recipient $\mathcal{R}$.

4 Proposed Schemes

In this section, we propose our 1-out-of-n oblivious signatures with n messages. In some applications, the message-set $\{m_1, \cdots, m_n\}$ (in which one of them will get signed) may be decided by the signer, while in other applications, they may be decided by the recipient. In either case, they should not affect the security of the scheme. In the following scheme, we assume that they are decided by the recipient.

4.1 Proposed 1-out-of-n Oblivious Signatures with n messages

System Setting: On input a security parameter 1^k, a signer $\mathcal{S}$ runs the System-Setup algorithm $\mathcal{G}$. The following output forms the public parameters of the scheme.

- p, q: two large primes such that $q|(p-1)$,
- g, h: two elements of $\mathbb{Z}_p^*$ of the same order q where the discrete logarithm $\log_g^h$ is unknown to all.
- $\mathcal{H} : \{0,1\}^* \rightarrow \mathbb{Z}_q^*$: an one way hash function.

Key Generation: $\mathcal{S}$ picks a random number $x \in \mathbb{Z}_q^*$ and computes $y \leftarrow g^x \bmod p$. x is kept secret as her private key and y is public as her public key.

Signature Generation: Assume that a recipient $\mathcal{R}$ would like to get a signature σ on a message $m_l \in \{m_1, \cdots, m_n\}$ which is obliviously signed by $\mathcal{S}$, then $\mathcal{R}$ executes the following protocol with $\mathcal{S}$:

Step 1 $\mathcal{R}$ starts the protocol by picking a random number $r \in \mathbb{Z}_q^*$, then computs $c = g^r h^l \bmod p$ and sends c together with the n messages $m_1, \cdots, m_n$ to the signer $\mathcal{S}$. Here l is the value of the subscript of m_l.

Step 2 For $i = 1, \cdots, n$, $\mathcal{S}$ picks a random number $k_i \in_R \mathbb{Z}_q^*$ and computes:

- $K_i \leftarrow g^{k_i} \bmod p$,
- $\hat{e}_i \leftarrow \mathcal{H}(m_i, K_i c/(gh)^i \bmod p)$, and
- $\hat{s}_i \leftarrow k_i - x\hat{e}_i \bmod q$.

$\mathcal{S}$ then sends $(\hat{e}_i, \hat{s}_i)$, $1 \le i \le n$, to $\mathcal{R}$.

step 3 For $1 \le i \le n$, $\mathcal{R}$ computes $\delta_i \leftarrow g^{(r-i)}h^{(l-i)} \bmod p$ and accepts the oblivious signature if and only if

$$\hat{e}_i = \mathcal{H}(m_i, g^{\hat{s}_i} y^{\hat{e}_i} \delta_i \bmod p) \quad 1 \le i \le n.$$

step 4 To convert the oblivious signature into a generic (Schnorr) signature, $\mathcal{R}$ computes:

- $e \leftarrow \hat{e}_l$, and
- $s \leftarrow r - l + \hat{s}_l \bmod q$,

The signature on m_l is $\sigma \leftarrow (e, s)$.

Signature Verification: Any verifier $\mathcal{V}$ accepts the signature σ as a valid signature on m_l if and only if

$$e = \mathcal{H}(m_l, g^s y^e \bmod p)$$

4.2 Security

Lemma 1. The proposed scheme is complete.

Proof. The completeness of the signature $\sigma = (e, s)$ is depended on the Schnorr signature. So, we only show the completeness of the oblivious signature $\sigma' = \left(\sum_{1 \le i \le n'} (\hat{e}_i, \hat{s}_i, \delta_i)\right)$ in Step 3 of the Signature Generation Algorithm. For any $(\hat{e}_i, \hat{s}_i, \delta_i)$ of $m_i \in \{m_1, \cdots, m_n\}$, we have

$$\begin{aligned}
&\mathcal{H}(m_i, g^{\hat{s}_i} y^{\hat{e}_i} \delta_i \bmod p) \\
&= \mathcal{H}(m_i, g^{k_i - x\hat{e}_i} y^{\hat{e}_i} g^{(r-i)} h^{(l-i)} \bmod p) \\
&= \mathcal{H}(m_i, g^{k_i - x\hat{e}_i} g^{x\hat{e}_i} g^{(r-i)} h^{(l-i)} \bmod p) \\
&= \mathcal{H}(m_i, g^{k_i} g^{(r-i)} h^{(l-i)} \bmod p) \\
&= \mathcal{H}(m_i, K_i g^r h^l/(gh)^i \bmod p) \\
&= \mathcal{H}(m_i, K_i c/(gh)^i \bmod p) \\
&= \hat{e}_i.
\end{aligned}$$

□

We than proof that the proposed scheme provides the security for signers.

Theorem 1. If there exists an adaptively chosen message attacker $\mathcal{B}$ which wins Game A with an advantage ϵ within a time T, then there exists an algorithm $\mathcal{A}$ which can solve the DL problem with the same advantage within a time $T' \leq 84480 q_h T/\epsilon$, where q_h is the number of hash queries.

Proof: $\mathcal{A}$ is given a DL problem (p, q, g, y) where p, q are two large primes such that $q|(p-1)$, and g, y are two elements of $\mathbb{Z}_p^*$ of the same order q. The purpose of $\mathcal{A}$ is to find $\log_g^y$, which is the solution to the DL problem.

In order to solve the problem, $\mathcal{A}$ utilizes $\mathcal{B}$ as a black-box. To get the black-box $\mathcal{B}$ run properly, $\mathcal{A}$ simulates the environments of the proposed $\mathcal{OS}_1^n$ scheme. In the following proof, we regard the hash function $\mathcal{H}$ as a random oracle. On the other hand, in the following proof, we assume that $\mathcal{B}$ is well-behaved in the sense that it always queries the random orale $\mathcal{H}$ on the message m^* that it outputs as its forgery. According to [2], we know that it is trivial to modify any adversary-algorithm $\mathcal{B}$ to have this property.

$\mathcal{A}$ picks a random number $h \in < g >$ and sets (p, q, g, h) as the system-wide parameters. Here $< g >$ denotes the subgroup of $\mathbb{Z}_p^*$ generated by g. In addition, $\mathcal{A}$ sets y as the signer's public key. $\mathcal{A}$ gives (p, q, g, h) and y to $\mathcal{B}$ and allows $\mathcal{B}$ to run via Game A.

In Game A, via an interactive way with $\mathcal{A}$, $\mathcal{B}$ takes part as a recipient in order to get a signature obliviously signed by the signing key $\log_g^y$. In order to respond for this query, for each $\mathcal{M}_i = \{m_{i_1}, \cdots, m_{i_n}\}$ and $c_i = g^{r_i} h^{l_i}$ chosen by $\mathcal{B}$, $\mathcal{A}$ does the following steps:

- For each $j = 1, \cdots, n$, $\mathcal{A}$ picks a random number $k_{i_j} \in_R \mathbb{Z}_q^*$ and
  - computes $K_{i_j} = g^{k_{i_j}} \bmod p$,
  - picks $\hat{s}_{i_j} \in_R \mathbb{Z}_q^*$,
  - sets $\hat{e}_{i_j} \leftarrow \mathcal{H}(m_{i_j}, K_{i_j} c_i/(gh)^j \bmod p)$.
- $\mathcal{A}$ returns $(\hat{e}_{i_j}, \hat{s}_{i_j}), 1 \leq i \leq n$, to $\mathcal{B}$ and records $(\mathcal{M}_i, c_i, K_{i_1}, \cdots, K_{i_n}, \hat{e}_{i_1}, \cdots, \hat{e}_{i_n}, \hat{s}_{i_1}, \cdots, \hat{s}_{i_n})$ to a Sign-List which is assumed to be initially empty.

The above execution can be executed at most t times. After the execution, $\mathcal{B}$ outputs its forgery $\sigma^* = (e^*, s^*)$ on a message m^*. Assume σ^* is a valid forgery and $\mathcal{B}$ wins Game A. According to the protocol, we have

$$e^* = \mathcal{H}(m^*, y^{e^*} g^{s^*} \bmod p).$$

Since $\mathcal{B}$ is assumed to be well-behaved, we have $e^* = \mathcal{H}(m_{i_j}, K_{i_j} c_i/(gh)^j \bmod p) = \hat{e}_{i_j}$ for some $\hat{e}_{i_j}$ and $m^* = m_{i_j}$ which are on the Sign-List.

We are now ready to apply the Forking Lemma [12]. By replaying the game with the same random tape but different choices of oracle $\mathcal{H}$, at the end of the second run, we obtain another valid forgery $(m^*, e^{*'}, s^{*'})$. Since $s^* = k_{i_j} + r_i - l_i + x_a e^*$ and $s^{*'} = k_{i_j} + r_i - l_i + x_a e^{*'}$ for the same $k_{i_j} + r_i - l_i$ (according to the Forking Lemma), we obtain $x_a = \frac{s^* - s^{*'}}{e^* - e^{*'}} \bmod q$. This is the solution to the DL problem. The advantage of $\mathcal{A}$ is the same as the advantage of $\mathcal{B}$ and the total running time T' of $\mathcal{A}$ is equal to the running time of the Forking Lemma [12] which is bound by $84480 q_h T/\epsilon$. Here q_h is the number of hash queries in the game. □

Theorem 2. The proposed scheme provides perfect security for recipients. In other words, the proposed scheme provides unconditional security on the ambiguity of the selected message.

Proof: It is sufficient to show that an attacker $\mathcal{F}$, taking parts as a signer, wins Game B with probability exactly the same as random guessing of $b \in \{0, 1\}$.

Assume $\mathcal{M} = \{m_1, \cdots, m_n\}$ and $c = g^r h^l$, $l \in \{1, \cdots, n\}$, where c is chosen by the recipient $\mathcal{R}$. It is easy to see that for any such c, there exists an $r_i \in \mathbb{Z}_q$ such that

$$c = g^r h^l = g^{r_1} h^1 = \cdots = g^{r_i} h^i = \cdots = g^{r_n} h^n \qquad \bmod p.$$

Consequently, we conclude that $\mathcal{F}$ wins Game B with probability exactly the same as random guessing of b. This ends the proof. □

5 Discussions and Performance Comparisons

Table 1. Performance Comparison 1

Scheme	Communication Cost			
	Numbers of Communication	Communication Over Head		
		$\mathcal{S} \to \mathcal{R}$ ($\approx bits$)	$\mathcal{R} \to \mathcal{S}$ ($\approx bits$)	$\mathcal{R} \to \mathcal{V}$ ($\approx bits$)
New	2	$2n\|q\|$ ($\approx 320n$)	$\|q\|$ (≈ 160)	$2\|q\|$ (≈ 320)
Chen [4]	3	$3n\|p\| + n\|q\|$ ($\approx 3232n$)	$\|q\|$ (≈ 160)	$7\|p\| + 2\|q\|$ (≈ 7488)
DSA-OT	2	$2n\|p\| + 2n\|q\|$ ($\approx 2368n$)	$\|p\|$ (≈ 1024)	$2\|q\|$ (≈ 320)

In Chen's schemes [4], the signature consists of two random signatures: $\sigma_{(g,y)}(m')$ and $\sigma_{(\mathcal{H}(m_l),m')}(m_l)$. $\sigma_{(g,y)}(m')$ is the signature on a

Table 2. Performance Comparison 2

Scheme	Computation Cost		
	Signer: $\mathcal{S}$	Receiver: $\mathcal{R}$	Verifier: $\mathcal{V}$
New	$2nEx.$	$(2n+2)Ex.$	$2Ex.$
Chen [4]	$3nEx.$	$(2n+10)Ex.$	$8Ex.$
DSA-OT	$5nEx.$	$3Ex.$	$2Ex.$

random message m' with secret key $x = \log_g^y$ and $\sigma_{(\mathcal{H}(m_l),m')}(m_l)$ is the signature on message m_l with the random secret key $e = \log_{\mathcal{H}(m_l)}^{m'}$. On the other hand, $\sigma_{(g,y)}(m')$ is generated by $\mathcal{S}$ in an interactive way with $\mathcal{R}$ but $\sigma_{(\mathcal{H}(m_l),m')}(m_l)$ is computed by $\mathcal{R}$ individually, so the oblivious property (from signer's viewpoint) is preserved. The verification of these two signatures assures anyone that the signer $\mathcal{S}$ with public key y indeed signed the message m_l.

Although, in Section 4, we only showed the construction of a 1-out-of-n oblivious signature with n messages, we emphasize that the proposed scheme can be easily modified into a 1-out-of-n oblivious signature with n keys. The technique is to use the signing key y_l in the proposed scheme instead of h. In this way, the signing key can be designated by a recipient. We omit the details and focus only on oblivious signature with n messages due to the space limitations.

In the following tables, we compare the efficiency of our scheme with that of Chen's scheme and the combination of the oblivious transfer (OT) scheme defined in [16] and DSA signature standard [10]. We denote Ex the exponentiation in $\mathbb{Z}_p$ and ignore other operations such as reversion and hash in all schemes.

In Table 1, we see that Chen's scheme requires three movies of communication between the signer $\mathcal{S}$ and the recipient $\mathcal{R}$ whereas our scheme and DSA-OT require only two movies of communication. In addition, our scheme enjoys small size of communication over head comparing with these two schemes. Table 2 shows the computation cost of each scheme and we can see that our scheme is more efficient than Chen's scheme in every step and more efficient than DSA-OT in the signing phase.

6 Conclusion

Oblivious signature is first proposed by Chen in 1994. However, previous references do not crisply formalize the notion, the model of the schemes as well as the security model. Moreover, previously proposed constructions

are less efficient in both communication and computation. In this paper, we give formal definitions on the model of oblivious signature schemes and give the security requirements of the scheme. We also propose a new oblivious signature scheme which is more efficient than Chen's scheme and the scheme based on DSA and oblivious transfer. Based on the discrete logarithm problem, the security of our scheme is proved in the random oracle model.

References

1. M. Abe and T. Okamoto, *Provably secure partially blind signatures*, Advances in Cryptology –CRYPTO'00, Lecture Notes in Computer Science **1880**, pp.271–286, 2000.
2. D. Boneh, B. Lynn and H. Shacham, *Short signatures from the Weil pairing*, Advances in cryptology –CRYPTO'01, Lecture Notes in Computer Science **2248**, pp.514–532, 2001.
3. D. Chaum, *Blind signatures for untraceable payments*, Advances in Cryptology –CRYPTO'82, Sringer-Verlag, pp.199–203, 1983.
4. L. Chen, *Oblivious signatures*, Proceedings of ESORICS'94, Lecture Notes in Computer Science **875**, pp.161–172, 1994.
5. D. Chaum and E. van Heijst, *Group signatures*, Advances in cryptology – EUROCRYPT'91, Lecture Notes in Computer Science **547**, pp.257–265, 1991.
6. L. Chen, T. Pedersen, *New group signature schemes*, Advances in cryptology – EUROCRYPT'94, Lecture Notes in Computer Science **950** pp.163–173, 1995.
7. A. Fiat and A. Shamir, *how to prove yourself: A randomized protocol for signing contracts*, Advances in cryptology –CRYPTO'86, Lecture Notes in Computer Science **263** pp.186–194, 1987.
8. S. Goldwasser, S. Micali and R. Rivest, *A digital signature scheme secure against adaptively chosen message atttacks*, SIAM Journal on Computing **17(2)** pp.281–308, 1988.
9. A. Juels, M. Luby and R. Ostrovsky, *Security of blind digital signatures*, Advances in cryptology –CRYPTO'97, Lecture Notes in Computer Science **1294** pp.150–164, 1997.
10. NIST (National Institute for Standard and Technology), Digital Signature Standard (DSS), FIPS PUB, **186**, 1994.
11. D. Pointcheval and J. Stern, *Security proofs for signature schemes*, Advances in cryptology –EUROCRYPT'96, Lecture Notes in Computer Science **1070** pp.387–398, 1996.
12. D. Pointcheval and J. Stern, *Security arguments for sigital signatures and blind signatures*, Journal of Cryptology, **13(3)**, pp.361–396, 2000.
13. M. O. Rabin, *How to exchange secrets by oblivious transfer*, Technical Report TR-81, Aiken Computation Laboratory, Harvard University, 1981. Also available at http://eprint.iacr.org/2005/187.pdf
14. C. P. Schnorr, *Efficient signature generation by smart cards*, Journal of Cryptology, **4(3)**, pp.161–174, 1991.
15. R. Tso, T. Okamoto and E. Okamoto, *Universal* 1*-out-of-n signatures*, In Proceedings of the Symposium on Cryptography and Information Security 2007 (SCIS'07), 2B4-3, 2007.

16. W. G. Tzeng, *Efficient* 1*-out-of-n oblivious transfer schemes with universal usable parameters*, IEEE Transactoins on Computer, **53(2)**, pp.232–240, 2004.