ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.18, No.3, March 2007, pp.746−754
DOI: 10.1360/jos180746

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

利用代理签名构造基于身份的优化公平交换协议*

徐 静[1+], 张振峰[2], 冯登国[2]

[1](信息安全国家重点实验室(中国科学院 研究生院),北京 100049)
[2](信息安全国家重点实验室(中国科学院 软件研究所),北京 100080)

Constructing Optimistic ID-Based Fair Exchange Protocols via Proxy Signature

XU Jing[1+], ZHANG Zhen-Feng[2], FENG Deng-Guo[2]

[1](State Key Laboratory of Information Security, Graduate School, The Chinese Academy of Sciences, Beijing 100049, China)
[2](State Key Laboratory of Information Security, Institute of Software, The Chinese Academy of Sciences, Beijing 100080, China)
+ Corresponding author: Phn: +86-10-88258713, Fax: +86-10-88258713, E-mail: xujing@is.iscas.ac.cn, http://www.is.ac.cn

Xu J, Zhang ZF, Feng DG. Constructing optimistic ID-based fair exchange protocols via proxy signature. *Journal of Software,* 2007,18(3):746−754. http://www.jos.org.cn/1000-9825/18/746.htm

Abstract: This paper introduces a natural paradigm for fair exchange protocols, called ID-based partial proxy signature scheme. A security model with precise and formal definitions is presented, and an efficient and provably secure partial proxy signature scheme is proposed. This is a full ID-based optimistic fair exchange protocol. Unlike the vast majority of previously proposed protocols, this approach does not use any zero knowledge proofs, and thus avoids most of the costly computations.

Key words: ID-based proxy signature; fair exchange protocol; provable security

摘 要: 为公平交换协议引入了一个自然的范例——基于身份的部分代理签名,给出其形式化的安全模型,同时提出了一个高效可证安全的部分代理签名方案.这是一个完全基于身份的优化公平交换协议.与以前协议不同的是,该方案没有使用任何零知识证明,有效地避免了大量计算.

关键词: 基于身份的代理签名;公平交换协议;可证明安全
中图法分类号: TP309 文献标识码: A

1 Introduction

With the growth of open networks such as Internet, the problem of fair exchange has become one of the fundamental problems in secure electronic transactions and digital rights management. Payment systems, contract signing, electronic commerce and certified e-mail are classical examples in which fairness is a relevant security property. Informally, an exchange protocol allows two distributed parties to exchange electronic data in an efficient and fair manner, and it is said to be fair if it ensures that during the exchange of items, no party involved in the

---

* Supported by the National Natural Science Foundation of China under Grant Nos.60373039, 90604010 (国家自然科学基金); the National Science Fund for Distinguished Young Scholars of China under Grant No.60025205 (国家杰出青年科学基金)

Received 2005-01-10; Accepted 2005-10-19

protocol can gain a significant advantage over the other party, even if the protocol is halted for any reason.

Significant effort has been devoted to the study of the fair exchange problem. Fair exchange protocols can be broadly categorized into three types:

(i) Gradual exchange protocols,

(ii) Protocols requiring an online trusted third party (TTP),

(iii) Protocols requiring an off-line TTP.

The first one is that two parties exchange data simultaneously. A simplified example to provide simultaneity is that they disclose the secret data bit by bit. This kind of scheme has a drawback that it requires many steps of interactions for exchanging data. In addition, one of these two parties will have an advantage of obtaining one more bit if he maliciously aborts in the middle of the protocol. The second approach is that an on-line TTP who acts as a mediator receives the data from both parties in each transaction and then forwards them to the accurate receivers[1]. However, TTP would become a bottleneck on communications since he takes part in all transactions, including the normal cases in which two parties honestly deliver their data. To improve the performance, optimistic fair exchange protocols based on an off-line TTP have been proposed. An optimistic fair exchange protocol usually involves three parties: users Alice and Bob, as well as an off-line TTP. The off-line TTP does not participate the actual exchange protocol in normal cases, and is invoked only in abnormal cases to dispute the arguments between Alice and Bob to ensure fairness.

Asokan, *et al*.[2] were the first to formally study the problem of optimistic fair exchanges. They presented several provably secure but highly interactive solutions, based on the concept of verifiable encryption of signatures. Their approach was later generalized by Ref.[3], but all these schemes involved expensive and highly interactive zero-knowledge proofs in the exchange phase. Other less formal works on interactive verifiably encrypted signatures include Refs.[4,5]. Ateniese[5] proposed six schemes for fair exchanges, while two of which were shown to be vulnerable to colluding attacks[6]. The first and only non-interactive verifiably encrypted signature scheme was constructed by Boneh, *et al*.[7], which is very elegant and provably secure in the random oracle model.

Shamir[8] firstly introduced the notion of identity-based (ID-based) cryptography in 1984. The main idea of ID-based cryptosystems is that the identity information of each user works as his/her public key, in other words, the user's public key can be calculated directly from his/her identity rather than being extracted from a certificate issued by a certificate authority (CA). Identity-based public key setting can be a good alternative for certificate-based public key setting, especially when efficient key management and moderate security are required.

But up to now, no one proposes an identity-based optimistic fair exchange protocol. Our current work is aimed at filling this void. Motivated by the approaches of verifiable probabilistic signatures[9] and verifiably committed signatures[10], we introduce a new paradigm for fair exchanges, called identity-based partial proxy signature. We present a formal model of ID-based partial proxy signatures, and propose an efficient and provably secure partial proxy signature scheme. As far as we know, the vast majority of fair exchange protocols require the use of zero knowledge proofs, which is the most computationally intensive part of the exchange protocol. Using proxy features of our model, we construct protocols that require no zero knowledge proofs in the exchange phase, and TTP does not need to maintain partial private key of each user which can be used to resolve a dispute. This will greatly reduce the communication overhead and managing cost.

The rest of the paper is organized as follows. The next section contains some preliminaries used in our scheme. In Section 3, we present an ID-based partial proxy signature scheme and formally analyze its security. In Section 4, an optimistic fair exchange protocol based on the scheme is proposed. And we end with concluding remarks in Section 5.

2 Definitions

2.1 The bilinear pairing

Let G be a cyclic additive group generated by P, whose order is a prime q, and V be a cyclic multiplicative group of the same order. Let $e:G\times G\rightarrow V$ be a pairing which satisfies the following conditions:

1. Bilinearity: For any $P,Q,R\in G$, we have $e(P+Q,R)=e(P,R)e(Q,R)$ and $e(P,Q+R)=e(P,Q)e(P,R)$. In particular, for any $a,b\in Z_q$, $e(aP,bP)=e(P,P)^{ab}=e(P,abP)=e(abP,P)$.
2. Non-degeneracy: There exists $P,Q\in G$, such that $e(P,Q)\neq 1$.
3. Computability: There is an efficient algorithm to compute $e(P,Q)$ for all $P,Q\in G$.

The typical way of obtaining such pairings is by deriving them from the weil-pairing or the tate-pairing on an elliptic curve over a finite field.

2.2 Gap Diffie-Hellman (GDH) groups

Let G be a cyclic group of prime order q and P be a generator of G.

1. The decisional Diffie-Hellman (DDH) problem is to decide whether $c=ab$ in Z/qZ for given $P,aP,bP,cP\in G$. If so, (P,aP,bP,cP) is called a valid Diffie-Hellman (DH) tuple.
2. The computational Diffie-Hellman (CDH) problem is to compute abP for given $P,aP,bP\in G$.

Now we present a definition for a gap Diffie-Hellman(GDH) group.

Definition 1. A group G is a gap Diffie-Hellman (GDH) group if the decisional Diffie-Hellman problem in G can be efficiently computable and there exists no efficient algorithm breaking computational Diffie-Hellman on G.

If we have an admissible bilinear pairing e in G, we can solve the DDH problem in G efficiently as follows: (P,aP,bP,cP) is a valid DH tuple $\Leftrightarrow e(aP,bP)=e(P,cP)$.

Hence an elliptic curve becomes an instance of a GDH group if the Weil (or the Tate) pairing is efficiently computable and the CDH is sufficiently hard on the curve.

2.3 ID-Based setting from bilinear pairings

The ID-based public key systems allow some public information of the user such as name, address and email *etc.*, rather than an arbitrary string to be used as his public key. The private key of the user is calculated by a trusted party, called PKG and sent to the user via a secure channel.

ID-based public key setting from bilinear pairings can be implemented as follows:

Let G be a cyclic additive group generated by P, whose order is a prime q, and V be a cyclic multiplicative group of the same order. A bilinear pairing is the map $e:G\times G\rightarrow V$. Define cryptographic hash function $H:\{0,1\}^*\rightarrow G$.

- ***g***: PKG chooses a random number $s\in Z_q^*$ and sets $P_{pub}=sP$. He publishes system parameters $params=\{G,V,e,q,P,P_{pub},H\}$ and keeps s secretly as the *master-key*.
- ***k***: A user submits his/her identity information ID and authenticates him to PKG. PKG computes the user's private key $d_{ID}=sQ_{ID}=sH(ID)$ and sends it to the user via a secure channel.

2.4 Proxy signature

The basic idea of most existing proxy signature schemes is as follows. The original signer sends a specific message with its signature to the proxy signer, who then uses this information to construct a proxy private key. With private key, the proxy signer can generate proxy signatures by employing a specified standard signature scheme. When a proxy signature is given, a verifier first computes the proxy public key from some public information, and then checks its validity according to the corresponding standard signature verification procedure.

A secure proxy signature scheme should satisfy the following four requirements[11]:

Verifiability: From the proxy signature, a verifier can be convinced of the original signer's agreement on the signed message.

Strong unforgeability: Only the designated proxy signer can create a valid proxy signature on behalf of the original signer. In other words, the original signer and other third parties who are not designated as a proxy signer cannot create a valid proxy signature. So it should also satisfy strong undeniability: Once a proxy signer creates a valid proxy signature on behalf of an original signer, he cannot repudiate the signature creation against anyone else.

Strong identifiability: Anyone can determine the identity of the corresponding proxy signer from a proxy signature.

Prevention of misuse: The proxy signer cannot use the proxy key for purposes other than generating a valid proxy signature. In case of misuse, the responsibility of the proxy signer should be determined explicitly.

3 ID-Based Partial Proxy Signatures

In the following, we would like to present an ID-based partial proxy signature scheme, and explicitly consider the attack models and security goals, which results in a concrete description for the security against all parties involved in the protocols.

3.1 ID-Based partial proxy signature scheme

We shall present an ID-based partial proxy signature scheme based on the standard ID-based proxy signature scheme[12]. An ID-based partial proxy signature scheme involves three entities: a signer Alice, a verifier Bob and an arbitrator TTP. As usual, let k be a security parameter, G be a GDH group of prime order $q>2^k$ generated by P, and $e{:}G\times G\rightarrow V$ is a bilinear map. Choose hash functions $H_1,H_2,H_3{:}\{0,1\}^*\rightarrow G$, and hash function $H_4{:}\{0,1\}^*\rightarrow Z_q^*$.

Setup: PKG picks a random master key $s\in Z_q^*$ and set $P_{pub}=sP$. TTP randomly chooses $s'\in Z_q^*$ and sets $P'=s'P$. TTP publishes $TPK=P'$ as a system parameter, and keeps $TSK=s'$ secret. Given Alice's identity ID_A and TTP's identity ID_T, PKG computes corresponding private key $d_A=sH_1(ID_A)$ and $d_T=sH_1(ID_T\|P')$.

TTP generates a warrant ω on messge m_ω to Alice as follows. The message m_ω contains the identity (ID) of the designated proxy signer Alice and, possibly, restrictions on the message the proxy signer is allowed to sign.

1. Randomly pick $r_\omega\in Z_q^*$ and compute $U_\omega=r_\omega P\in G$ and then put $H_\omega=H_2(ID_T,m_\omega,U_\omega)\in G$.
2. Compute $V_\omega=d_T+r_\omega H_\omega\in G$.

The signature on m_ω is the warrant $\omega=\langle U_\omega,V_\omega\rangle$.

Sig and Psig: At first Alice verifies signature $\omega=\langle U_\omega,V_\omega\rangle$ by $e(P,V_\omega)=e(P_{pub},H_1(ID_T\|P'))e(U_\omega,H_\omega)$, here $H_\omega=H_2(ID_T,m_\omega,U_\omega)$. Alice computes proxy signature and partial proxy signature on message m_ω as follows.

1. Randomly pick $r_p\in Z_q^*$ and compute $U_p=r_pP\in G$ and then put $H_p=H_3(ID_A,m,U_p)\in G$.
2. Compute $V_p=H_4(ID_T,ID_A,m_\omega,U_\omega)d_A+V_\omega+r_pH_p\in G$ and $V_p'=H_4(ID_T,ID_A,m_\omega,U_\omega)d_A+V_\omega+r_pH_p+r_pP'\in G$.

The proxy signature and partial proxy signature on m is

$$\sigma=Sig(m,ID_T,ID_A,d_A,\omega)=(U_p,V_p,m_\omega,U_\omega)$$

and

$$\sigma'=Psig(m,ID_T,ID_A,d_A,\omega,P')=(U_p,V_p',m_\omega,U_\omega),$$

respectively.

Ver and Pver: To verify a proxy signature $\sigma=(U_p,V_p,m_\omega,U_\omega)$ on message m, the algorithm **Ver** checks

$$e(P,V_p)=e(P_{pub},H_1(ID_A))^{H_4(ID_T,ID_A,m_\omega,U_\omega)}e(P_{pub},H_1(ID_T\|P'))e(U_p,V_p)e(U_\omega,H_\omega) \qquad (1)$$

To verify a partial proxy signature $\sigma'=(U_p,V_p',m_\omega,U_\omega)$ on message m, the algorithm **Pver** checks

$$e(P,V_p')=e(P_{pub},H_1(ID_A))^{H_4(ID_T,ID_A,m_\omega,U_\omega)}e(P_{pub},H_1(ID_T\|P'))e(U_p,H_p+P')e(U_\omega,H_\omega) \qquad (2)$$

where $H_p=H_3(ID_A,m,U_p)\in G$ and $H_\omega=H_2(ID_T,m_\omega,U_\omega)\in G$.

Res: Given a partial proxy signature $\sigma'=(U_p,V_p',m_\omega,U_\omega)$ on message m, the arbitrator TTP first verifies its validity by checking Eq.(2). If valid, TTP computes $V_p=V_p'-s'U_p$ and returns $\sigma=(U_p,V_p,m_\omega,U_\omega)$ as a proxy signature of m to the verifier.

Remark:

(1) Recall that in a verifiable committed signature scheme[10] and most of the verifiable encrypted signature schemes, TTP shall maintain a secret-public key pair for each user via a registration phase, and the secret keys will then be used to resolve a dispute. In our partial proxy signature scheme, TTP only needs to publish a public system parameter and generate a warrant ω. No further registration is needed and no zero-knowledge proofs are involved, which will greatly reduce the communication overhead and managing cost.

(2) In the Setup phase, the private key of TTP is computed by its identity and public parameter TPK, which efficiently prevents the adversary from changing TPK.

(3) In our partial proxy signature scheme, the standard ID-based proxy signature scheme can be replaced by any other secure proxy signature scheme.

Correctness: The correctness of an ID-based partial proxy signature scheme states that

- $Ver(m,Sig(m,ID_T,ID_A,d_A,\omega),ID_T,ID_A,TPK)=1$
- $Pver(m,Psig(m,ID_T,ID_A,d_A,\omega,TPK),ID_T,ID_A,TPK)=1$
- $Ver(m,Res(m,\sigma',TSK),ID_T,ID_A,TPK)=1$

The correctness of the above scheme is obvious.

3.2 Security of ID-based partial proxy signatures

The security of ID-based partial proxy signatures consists of ensuring three aspects: security against signer Alice, security against verifier Bob, and security against arbitrator TTP. In the following, we denote by $\boldsymbol{O_{Psig}}$ an oracle simulating partial proxy signing procedure, $\boldsymbol{O_{Res}}$ an oracle simulating the resolution procedure, and $\boldsymbol{O_{Ext}}$ an oracle simulating private key extraction procedure. Let k be a security parameter, and PPT stand for "probabilistic polynomial time" (in the security parameter).

Security against a signer. We require that any PPT adversary $\mathcal{A}$ succeeds with at most negligible probability in the following experiment:

$$Setup^*(1^k)\rightarrow(d_A^*,TPK,TSK);\ (m,\sigma')\leftarrow A^{O_{Res},O_{ext}}(d_A^*,TPK)\ ;\ \sigma\leftarrow Res(m,\sigma',TSK)$$

Success of $\mathcal{A}=[Pver(m,\sigma',ID_T,ID_A,TPK)=1\wedge Ver(m,\sigma,ID_T,ID_A,TPK)=0]$
where $Setup^*$ denotes the run of $Setup$ with dishonest Alice (run by the adversary $\mathcal{A}$) and d_A^* is $\mathcal{A}$'s state after this run. In other words, Alice should not be able to produce partial signature σ' which looks good to Bob, but which will not be opened into Alice's full signature by the honest TTP.

Security against a verifier. Intuitively, a verifier Bob should not be able to transfer any of partial proxy signatures σ' that he got from Alice into a proxy signature σ, without explicitly asking TTP to do that. More precisely, we require that any PPT adversary $\mathcal{A}$ succeeds with at most negligible probability in the following experiment:

$$Setup^*(1^k)\rightarrow(d_A^*,TPK,TSK);\ (m,\sigma)\leftarrow A^{O_{Psig},O_{Res},O_{ext}}(d_A^*,TPK)$$

Success of $\mathcal{A}=[Ver(m,\sigma,ID_T,ID_A,TPK)=1\wedge ID_A\notin Query(A,O_{Ext})\wedge(m,\sigma')\notin Query(A,O_{Res})]$.

where $Query(A,O_{Ext})$ is the set of valid queries $\mathcal{A}$ asked to the private key extraction oracle O_{Ext}, and $Query(A,O_{Res})$ is the set of valid queries $\mathcal{A}$ asked to the resolution oracle O_{Res}, i.e., the set of (m,σ') the adversary $\mathcal{A}$ queried to O_{Res} satisfying $Pver(m,\sigma',TPK)=1$.

Security against an arbitrator. This property is crucial. Even though the arbitrator TTP is semi-trusted, the primary signer Alice does not want TTP to produce a valid proxy signature which she did not intend on producing. To achieve this goal, we require that any PPT adversary $\mathcal{A}$ associated with partial proxy signing oracle $\boldsymbol{O_{Psig}}$ and private key extraction oracle $\boldsymbol{O_{Ext}}$, succeeds with at most negligible probability in the following experiment:

$$Setup^*(1^k)\rightarrow(d_A,TSK^*,TPK);\ (m,\sigma)\leftarrow A^{O_{Psig},O_{ext}}(TSK^*,TPK)$$

Success of $\mathcal{A}$=$[Ver(m,\sigma,ID_T,ID_A,TPK)=1\wedge ID_A\notin Query(A,O_{Ext})\wedge m\notin Query(A,O_{Psig})]$.

where $Setup^*$ denotes the run of $Setup$ with the dishonest arbitrator $\mathcal{A}$, and TSK^* is her state after this run, and $Query(A,O_{Psig})$ is the set of queries $\mathcal{A}$ asked to the partial proxy signing oracle $\boldsymbol{O_{Psig}}$.

Definition 2. An ID-based partial proxy signature scheme is secure if it is secure against the signer, the verifier and the arbitrator.

Theorem 1. The ID-based partial proxy signature scheme above is secure in GDH groups.

Proof: Note that the underlying ID-based proxy signature scheme **Sig** is secure against forgery in GDH groups[12]. Similarly we can show that ID-based partial proxy signature scheme **Psig** is also secure against forgery in GDH groups.

According to Definition 2, we shall show that the proposed partial proxy signature scheme is secure against signer, verifier and arbitrator.

Secure against signer's attack: For a malicious signer, with the help of the oracle $\boldsymbol{O_{Res}}$ and $\boldsymbol{O_{Ext}}$, her goal is to produce a valid partial proxy signature $\sigma'=(U_p,V'_p,m_\omega,U_\omega)$ on message m, which cannot be extracted into a valid proxy signature $\sigma=(U_p,V_p,m_\omega,U_\omega)$. However, this is always not the case. Any valid partial signature σ' satisfies Eq.(2), so the resolved full signature must satisfy Eq.(1) according to $V_p=V_p^{\tilde{\prime}}s'U_p$. In fact, the oracle $\boldsymbol{O_{Res}}$ cannot give any help to a malicious signer: she has already known what $\boldsymbol{O_{Res}}$ extracted.

Secure against verifier's attack: An adversarial verifier's goal, making use of oracles $\boldsymbol{O_{Psig}}$, $\boldsymbol{O_{Ext}}$ and $\boldsymbol{O_{Res}}$, is to forge a valid proxy signature σ, for which the corresponding partial proxy signature σ' has not been queried to $\boldsymbol{O_{Res}}$. Suppose adversary verifier B is successful in such an attack, we show how to construct an algorithm Φ that solves CDH problem in G. This will contradict the fact that G is GDH group.

Algorithm Φ is given $X=xP\in G$ and $Y=yP\in G$. Its goal is to output $xY=xyP\in G$. Algorithm Φ simulates the challenger and interacts with adversary B as follows.

Φ picks randomly $P_{pub}\in G$, and initializes B with $(P,P_{pub},P'=X)$ as a system parameter.

To respond to the random oracle H_1 queries, Φ maintains a list L_1 of tuples $\langle ID_i,b_i\rangle$ as explained below. The list is initially empty. When an identity ID is submitted to the oracle H_1, algorithm Φ responds as follows:

1. If the query ID already appears on the list L_1 in some tuple $\langle ID,b\rangle$, then algorithm Φ responds with $H_1(ID)=bP$.

2. Otherwise, algorithm Φ picks $b\in Z_q^*$ at random, stores the tuple $\langle ID,b\rangle$ in the list L_1 and returns bP as a hash value to the adversary B.

For other random oracle queries, Φ makes similar answers.

When B requests the private key associated to an identity ID_i, Φ recovers the corresponding $\langle ID_i,b_i\rangle$ from L_1. It means that $H_1(ID_i)$ was previously defined to be b_iP and b_iP_{pub} is then returned to B as a private key associated to ID_i.

For an $\boldsymbol{O_{Psig}}$ query on message m_i, Algorithm Φ responds to this query as follows.

- Recover the previously defined value $Q_T=H_1(ID_T\|P')$ and $Q_A=H_1(ID_A)$ from the list L_1.
- Pick $t_1,t_2\in Z_q^*$ at random and define $V_p'=t_1P_{pub},U_\omega=t_2P_{pub}$.
- Make query $(ID_T,ID_A,m_\omega,U_\omega)$ to H_4 oracle and return $H_4(ID_T,ID_A,m_\omega,U_\omega)=\mu$.
- Similar to Ref.[13], Φ generates a random coin $c\in\{0,1\}$ such that $\Pr[c=0]=1/q_{ps}$. Picks a random $r\in Z_q^*$, Φ define $U_p=crP+(\tilde{1}c)Y$. Here the adversary B makes at most q_{ps} queries to $\boldsymbol{O_{Psig}}$.
- Pick $t_3\in Z_q^*$ at random and define the hash value $H_3(ID_A,m,U_p)$ as $t_3P_{pu\beta}-P'$ (Φ output "failure" and halts if H_3 turns out to be already defined for the input $\langle ID_A,m,U_p\rangle$). Define the hash value $H_2(ID_T, m_\omega, U_\omega)$ as $t_2^{-1}(t_1P-Q_T-\mu Q_A-t_3U_p)$ (Φ output "failure" and halts if H_2 turns out to be already defined for the input $\langle ID_T,m_\omega,U_\omega\rangle$).
- $\sigma_i'=(U_p,V_p',m_\omega,U_\omega)$ is a valid partial proxy signature. If $U_p\neq Y$, Φ adds (m_i,σ_i',r_i) to a list L.

To simulate a valid $\boldsymbol{O_{Res}}$ query on (m',σ'), Φ just looks up the list L, answers Bob with $\sigma=(U_p,V_p^{\tilde{}}\, r_iP',m_\omega,U_\omega)$ if $(m',\sigma'=(U_p,V_p',m_\omega,U_\omega),r_i)$ is in the list, and halts otherwise.

Suppose Bob outputs a proxy signature forgery $\sigma^*=(U_p^*,V_p^*,m_\omega^*,U_\omega^*)$ in the ultimate, for which the corresponding partial proxy signature $\sigma'^*=(U_p^*,V_p'^*,m_\omega^*,U_\omega^*)$ has not been queried to $\boldsymbol{O_{Res}}$. From Eq.(1) and Eq.(2), we have $e(P,V_p^*-V_p'^*)=e(U_p^*,P')$. Φ declares failure and halts if $U_p^*\neq Y$. Otherwise, Φ calculates and outputs the required xY as $V_p^*-V_p'^*$. This completes the description of algorithm Φ.

Secure against arbitrator's attack: Now we consider an adversarial TTP's attack. We shall convert such an attack into a forger Φ against the underlying ID-based proxy signature scheme[12]. Note that Φ takes as input (P,P_{pub}) and has access to the signing oracle $\boldsymbol{O_{Sig}}$ and the private key extraction oracle $\boldsymbol{O_{Ext}}$ of the underlying ID-based proxy signature scheme. While TTP accepts (P,P_{pub},s',P') as inputs, and has access to oracles $\boldsymbol{O_{Psig}}$ and $\boldsymbol{O_{Ext}}$, and wins if he forges a valid proxy signature σ for some message m without making a query m to $\boldsymbol{O_{Psig}}$ and a query ID_A to $\boldsymbol{O_{Ext}}$.

So here is how Φ simulates the run of TTP. It picks a random $s'\in Z_q^*$, sets $P'=s'P$ and gives (P,P_{pub},s',P') to TTP. Φ can respond to $\boldsymbol{O_{Ext}}$ queries ID of TTP by getting the corresponding private key from its own extraction oracle. Φ can respond to $\boldsymbol{O_{Psig}}$ queries m of TTP by first getting a signature $\sigma=(U_p,V_p,m_\omega,U_\omega)$ from its own signing oracle, and then returning $\sigma'=(U_p,V_p',m_\omega,U_\omega)$, here $V_p'=V_p+s'U_p$. Finally when TTP outputs the forgery (m,σ), Φ also outputs the same forgery. We see that the simulation is perfect.

The above arguments show that, if an adversary can attack our partial proxy signature scheme, then one can solve CDH problem in G. □

4 Fair Exchanges Based on Partial Proxy Signature

Now we present an optimistic fair exchange protocol based on the partial proxy signature scheme described in Section 3.

Let G be a GDH group of prime order q generated by P. PKG picks a random master key $s\in Z_q^*$ and sets $P_{pub}=sP$. TTP randomly chooses $s'\in Z_q^*$ and sets $P'=s'P$. TTP publishes $TPK=P'$ as a system parameter, and keeps $TSK=s'$ secret. Given Alice's identity ID_A and TTP's identity ID_T, PKG computes the corresponding private key $d_A=sH_1(ID_A)$ and $d_T=sH_1(ID_T\|P')$. TTP generates a warrant ω to Alice.

1. With the warrant ω, Alice computes the partial proxy signature σ'_A and proxy signature σ_A on message m. Then Alice sends σ'_A to Bob.
2. Bob first checks σ'_A by Eq.(2). If it is valid, Bob sends his proxy signature σ_B to Alice.
3. After receiving Bob's proxy signature σ_B, Alice verifies σ_B by Eq.(1). If valid, she sends proxy signature σ_A to Bob.
4. If Bob does not receiving anything in Step 3, or if σ_A is invalid, then he sends Alice's partial proxy signature σ'_A and his own proxy signature σ_B to TTP. TTP first verifies the validity of σ'_A and σ_B. Then TTP computes $\sigma_A = Res(\sigma'_A, TSK)$. TTP sends σ_A to Bob and sends σ_B to Alice.

Security of the protocol follows directly from Theorem 1.

5 Conclusion

In this paper, we present a novel method for constructing efficient ID-based optimistic fair exchange protocols using partial proxy signature. We introduce a formal definition of partial proxy signature and propose an efficient and provably secure partial proxy signature scheme. The resulting optimistic fair exchange protocol does not involve zero knowledge proofs in the exchange phase, and TTP does not maintain partial private key of each user which can be used to resolve a dispute, which greatly reduces the communication overhead and managing cost. This is the first efficient ID-based optimistic fair exchange protocol.

References:

[1] Zhou J, Gollmann D. A fair non-repudiation protocol. In: Proc. of the IEEE Symp. on Security and Privacy. Oakland: IEEE Computer Press, 1996. 55−61.

[2] Asokan N, Shoup V, Waidner M. Optimistic fair exchange of digital signatures. IEEE Journal on Selected Areas in Communication, 2000,18(4):593−610.

[3] Camenisch J, Damgard IB. Verifiable encryption, group encryption, and their applications to group signatures and signature sharing schemes. In: Okamoto T, ed. Proc. of the Advances in Cryptology—ASIACRYPT 2000. LNCS 1976, Berlin, Heidelberg: Springer-Verlag, 2000. 331−345.

[4] Bao F, Deng RH, Mao W. Efficient and practical fair exchange protocols with off-line TTP. In: Proc. of the IEEE Symp. on Security and Privacy. Oakland: IEEE Computer Press, 1998. 77−85.

[5] Ateniese G. Verifiable encryption of digital signatures and applications. ACM Trans. on Information and System Security, 2004, 7(1):1−20.

[6] Bao F. Colluding attacks to a payment protocol and two signature exchange schemes. In: Lee PJ, ed. Proc. of the Advances in Cryptology—ASIACRYPT 2004. LNCS 3329, Berlin, Heidelberg: Springer-Verlag, 2004. 417−429.

[7] Boneh D, Gentry C, Lynn B, Shacham H. Aggregate and verifiably encrypted signatures from bilinear maps. In: Biham E, ed. Proc. of the Advances in Cryptology—EUROCRYPT 2003. LNCS 2656, Berlin, Heidelberg: Springer-Verlag, 2003. 416−432.

[8] Shamir A. Identity-Based cryptosystems and signature schemes. In: Blakley GR, ed. Proc. of the Advances in Cryptology—Crypto'84. LNCS 196, Berlin, Heidelberg: Springer-Verlag, 1984. 47−53.

[9] Zhang ZF, Zhou YB, Feng DG. Efficient and optimistic fair exchange based on standard RSA with provable security. IACR Cryptology ePrint Archive, Report 2004/351, 2004.

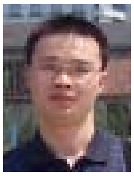
[10] Dodis Y, Reyzin L. Breaking and repairing optimistic fair exchange from PODC 2003. In: Proc. of the ACM Workshop on Digital Rights Management (DRM). New York: ACM Press, 2003. 47−54.

[11] Lee JY, Cheon JH, Kim S. An analysis of proxy signatures: Is a secure channel necessary? In: Joye M, ed. Topics in Cryptology—CT-RSA, The Cryptographers' Track at the RSA Conference 2003. LNCS2612, Berlin, Heidelberg: Springer-Verlag, 2003. 68−79.

[12] Xu J, Zhang ZF, Feng DG. ID-Based proxy signature using bilinear pairings. In: Chen G, ed. Parallel and Distributed Processing and applications—ISPA 2005 Workshops. LNCS 3759, Berlin, Heidelberg: Springer-Verlag, 2005. 359−367.

[13] Bellare M, Rogaway P. Random oracles are practical: A paradigm for designing efficient protocols. In: Proc. of the 1st ACM Conf. on Computer and Communications Security. New York: ACM Press, 1993. 62−73.

XU Jing was born in 1972. She is an associate professor at the State Key Laboratory of Information Security, Graduate School of the Chinese Academy of Sciences. Her research areas are design and analysis of security protocols.



ZHANG Zhen-Feng was born in 1972. He is an associate professor at the Institute of Software, the Chinese Academy of Sciences, and a CCF senior member. His research areas are theoretical and applied cryptography, design and analysis of security protocols, theory and technology of information security.

FENG Deng-Guo was born in 1965. He is a professor and doctoral supervisor at the Institute of Software, the Chinese Academy of Sciences, and a CCF senior member. His research areas are information security and network security.

第 7 届全国虚拟现实与可视化学术会议(CCVRV 2007)

征文通知

由中国计算机学会虚拟现实与可视化技术专业委员会、中国图像图形学会虚拟现实与可视化技术专业委员会和中国系统仿真学会虚拟现实技术专业委员会主办，北京航空航天大学承办的第 7 届全国虚拟现实与可视化技术及应用学术会议将于 2007 年 10 月在北京举行。本次会议将集聚国内从事虚拟现实与可视化技术的研究人员和工程技术人员，广泛开展学术交流、研究发展战略、推动成果转化、共同促进虚拟现实与可视化技术的发展与应用。

本次大会录用的学术论文将在核心期刊《系统仿真学报》(增刊)发表。会议将邀请国内外著名专家作专题报告，同时将举办科研成果和最新产品展示会，为各研究开发单位及有关厂商展示自己的成果、产品提供场所。欢迎大家积极投稿。

一、征文范围(包括但不限于)

建模技术、动画技术、可视化技术、多媒体技术、人机交互技术、虚拟制造、仿真技术、分布式系统、空间化声音、模式识别应用、图形平台、网络技术、遥操作技术、VRML 技术、逼真图形图像技术、增强现实、协同操作、数字博物馆、网络游戏、图象绘制技术、可视化地理信息系统、基于图像的视景生成技术、虚拟现实与可视化应用系统……

二、征文要求

1、论文未被其他会议、期刊录用或发表，不超过 10 页；2、要求接受电子投稿(同时提交 Word 与 Pdf 格式文件)；3、论文包含：题目、中英文摘要、正文、参考文献等；4、正式论文格式见论文录用通知；5、投稿者请在论文最后务必写清姓名、单位、通信地址、电话及 E-mail 地址。

三、重要日期

征文截止日期：2007 年 6 月 15 日(收到日期)

录用通知日期：2007 年 7 月 15 日(发出日期)

四、来稿联系方式(请注明 CCVRV07 征文)

联系单位：北京航空航天大学 6863 信箱(邮政编码：100083)

联 系 人：伍潇潇，胡勇，范志强，王正光 电话：13161965259

电子邮件: ccvrv07@vrlab.buaa.edu.cn

五、会议网站

http://vrlab.buaa.edu.cn 欢迎上网查询大会各项文件和最新通知