Anonymous Signature Schemes

Guomin Yang[1], Duncan S. Wong[1], Xiaotie Deng[1], and Huaxiong Wang[2]

[1] Department of Computer Science
City University of Hong Kong
Hong Kong, China
{csyanggm,duncan,deng}@cs.cityu.edu.hk
[2] Department of Computing
Macquarie University
Australia
hwang@ics.mq.edu.au

November 15, 2005

Abstract. Digital signature is one of the most important primitives in public key cryptography. It provides authenticity, integrity and non-repudiation to many kinds of applications. On signer privacy however, it is generally unclear or suspicious of whether a signature scheme itself can guarantee the anonymity of the signer. In this paper, we give some affirmative answers to it. We formally define the signer anonymity for digital signature and propose some schemes of this type. We show that a signer anonymous signature scheme can be very useful by proposing a new anonymous key exchange protocol which allows a client Alice to establish a session key with a server Bob securely while keeping her identity secret from eavesdroppers. In the protocol, the anonymity of Alice is already maintained when Alice sends her signature to Bob in clear, and no additional encapsulation or mechanism is needed for the signature. We also propose a method of using anonymous signature to solve the collusion problem between organizers and reviewers of an anonymous paper review system.

Keywords: Anonymity, Digital Signature, Key-Privacy

Table of Contents

1 Introduction

Digital signature is one of the most important primitives in public key cryptography. It is a very useful tool for providing authenticity, integrity and non-repudiation while it has seldom been considered to provide user privacy by its own. In many applications such as e-voting, e-auction, authentication protocols, and many others, we need to protect a signer's identity from being known by eavesdroppers or other parties in a system. For example, in an anonymous electronic transaction processing system [17] or an anonymous key exchange protocol [27], additional mechanisms or encapsulation techniques such as extra layers of encryption are applied onto their underlying signature schemes for protecting the signer's identity. In some other examples such as [8], several requirements for the signer anonymity of a signature scheme are informally given. However, among these solutions or discussions, they usually require significant increase of system complexities or lack formal methodologies for analyzing the level of anonymity being provided to signers. Although it is widely believed that a signature scheme by itself may provide a certain degree of anonymity to its signers, there is no formal treatment on this subject. It is still generally unclear on exactly what conditions that a signature itself can provide anonymity of its signer. Comparing with the progress on the decryptor identity exposure issue of public key encryption schemes [2], it has been far lagged behind on the research of the signer anonymity of signature schemes themselves.

In this paper, we formally define *a signer anonymous signature scheme*. Traditionally, a secure digital signature scheme is required to be existentially unforgeable against adaptive chosen message attack (euf-cma) [15]. By signer anonymity, we mean that given a signature (but not the message), no one can tell who the real signer is. It will coexist with unforgeability. That is, a signer anonymous signature scheme will be euf-cma as a conventional signature scheme, while the signer's identity will be protected if only a signature is given but not the corresponding message or signer's identity/public key. We are expecting to see the signer anonymous signature schemes to be very useful for many privacy-related applications. In particular, we will see that it is possible to just use signer anonymous signatures to preserve user privacy without applying any additional mechanism or encapsulation technique onto the signatures.

One may think that keeping the message of a signature secret should give signer anonymity to the signature. However, this is far from enough. Also notice that a system may only have a few public keys so that an adversary can efficiently enumerate them all in an endeavor of finding signer's identity. Due to the lack of a rigorous treatment on the signer anonymity of a signature scheme, signatures used in many current privacy-related systems are giving out enough information to an adversary for compromising the identity of an anonymous user.

Consider the following example (Fig. 1) which is a key transport protocol proposed by Boyd and Park [8] for a mobile client A to transport a session key σ to a server B. The protocol is also targeted to provide client anonymity by protecting A's identity ID_A from being known by eavesdroppers.

$$\begin{aligned}
&A \rightarrow B : PKE_B(ID_A,\ \sigma,\ count)\\
&A \leftarrow B : Enc_\sigma(count,\ r_B)\\
&A \rightarrow B : Sig_A(ID_B,\ h(count, \sigma, r_B))
\end{aligned}$$

Fig. 1. Boyd-Park Authenticated Key Transport Protocol

In the first message of the protocol, A encrypts ID_A, σ and a field *count* under B's public key encryption function PKE_B which is assumed to be publicly known. This protects A's identity from being known by eavesdroppers. In the third message of the protocol however, A also needs to generate and send a signature to B in clear. Obviously, to hide the identity of A, this signature should not provide any meaningful information about A's identity to eavesdroppers.

To illustrate some subtleties of making a signature signer anonymous, we describe several potential attacking techniques which can be used to compromise a signer's identity. They are

- Different Domain Attack
- Redundant Structure Attack

- Sparse Message Attack

Redundant Structure Attack. As remarked by the authors in [8], it is important to make sure that the signature does not contain any "redundant" structure, which can be revealed during the signature verification procedure and does not require the signed message to be known, while such a redundant structure may help an eavesdropper identify the mobile client. In [8], no formal definition of such "redundant" structure is given and it is also not obvious to have a satisfactory definition for it. We may only give some examples to illustrate the idea of it. For example, a *recoverable* signature scheme [7] allows the message to be recovered and verified from the redundant structure of such a signature once the correct signature verification function is given. Hence if the signature scheme Sig_A in the protocol above is recoverable, an eavesdropper can find out the identity of A by trying the signature verification functions of all mobile clients one by one until a message starting with ID_B is recovered and verified.

Different Domain Attack. In order to prevent Redundant Structure Attack, a signature scheme which appears to be immune from such an attack, an ElGamal or Schnorr [25] type signature scheme was chosen for this key transport protocol [8]. However, we notice that an eavesdropper may still be able to identify the mobile client by examining the signature from another aspect: simply from *the length of a signature*. Suppose there are two mobile clients in the system and one of them is communicating with the server using this anonymous key transport protocol. When Schnorr signature scheme is used, the two mobile clients may select their own keys in different groups that could have different sizes. By examining the length of the signature in the protocol, the eavesdropper can tell which mobile client is communicating with the server.

Sparse Message Attack. For signature schemes where redundant structure does not exist and all signers have the same signature domain, an adversary may still be able to find out the signer from just the given signature. Below is an example.

Consider a trapdoor one-way permutation family indexed by signers' public keys (e.g. RSA [23]), a signature of a message is generated by computing the permutation inverse of the message using a signer's private signing key (i.e. a trapdoor information). If the message space is sparse in the image of the permutation family (e.g. the image of the permutation family contains only a few meaningful messages), the adversary is able to find out who the actual signer is. Given a signature, the adversary can find out the actual signer's identity using the following elimination method: "For a trial signer, the adversary computes the one-way permutation of the signature indexed by the signer's public key and checks if the result is in the corresponding message space. If it is not, then the adversary is sure that this signer is not the actual signer of the signature. The adversary will simply repeat this elimination procedure until a signer is found."

There are many other examples that signature schemes may have leaked too much information about the identity of the signer. In [20], Lee et al. used a signature scheme for anonymous electronic auction, and the signature scheme is very similar to Schnorr signature scheme. In the scheme, if bidders are allowed to pick keys in different cyclic groups that are arbitrarily chosen, then it is possible that keys are of different lengths. The method of examining the length of signatures can usually give enough information to eavesdroppers for finding out the identities of bidders.

Encrypting a Signature May Not Work Either. In [17], a pseudonym server is used to enhance user privacy in electronic transactions (e.g. SET [21], iKP [3]). Although signatures exchanged between a client and a pseudonym server are encrypted, client identity could still be revealed from information such as the length of a ciphertext which is usually tightly related to the length of the signature encrypted. One should note that this problem may not be solved by using a key-privacy-enabled public key encryption scheme [2] as such an encryption scheme is addressing the identity exposure issue of the decryptor, not the sender.

Contributions. We formally introduce signer anonymous digital signature and define two security models subsequently for it. The first one is *static*, it provides an intuitive way to screen off signatures which do not have the anonymity property; the second one, a stronger model, combines the static model with the adaptive chosen message attack, and this *adaptive* model is then used in the security analyses of the signer anonymity of our proposed schemes.

Some commonly used signature schemes are examined. We show that the basic RSA signature scheme [23] is in general not signer anonymous, except in a special case where some restrictive assumptions are applied. We then show that PSS [7] is not signer anonymous even with those restrictive assumptions. We also show that Schnorr and ElGamal signature schemes are not signer anonymous, except all signers are choosing keys under a common domain.

To transform those signature schemes to signer anonymous versions, we propose some extensions of them and show that they are signer anonymous even under our adaptive model. We also propose a new anonymous key exchange protocol which allows a client Alice to establish a session key with a server Bob securely while keeping her identity secret from eavesdroppers. In the protocol Alice sends her signer anonymous signature to Bob in clear, while the anonymity of Alice is already maintained. As another application, we propose a method of using anonymous signature to solve the collusion problem between organizers and reviewers of an anonymous paper review system.

Paper Organization. In Sec. 2, we review some related work. This is followed by Sec. 3 in which we introduce a security model for signer anonymous signature. In Sec. 4, we review some commonly used signature schemes and show that they are not signer anonymous. In Sec. 5, we introduce a stronger model for signer anonymous signature and call it the adaptive model. In Sec. 6, we propose some modifications of the signature schemes reviewed in Sec. 4 and show their anonymity under the stronger adaptive model. In Sec. 7, we apply our anonymous signature schemes on the design of anonymous key establishment protocols and the construction of an anonymous paper review system which solves the collusion problem between organizers and reviewers.

2 Related Work

For the counterpart of digital signature in public key cryptography, the public key encryption with key privacy was introduced and first formalized by Bellare et al. in [2]. In their model, a secure key-privacy-enabled encryption scheme not only ensures that an encrypted message is semantically secure against adaptive chosen-ciphertext attacks but also prevents the public from getting the decryptor's identity from his ciphertexts. Several techniques were also proposed in [2] for converting a conventional encryption scheme to a key-privacy-enabled encryption scheme. However, these techniques cannot be simply applied to digital signature schemes for converting them to anonymous version. The main challenge of constructing an anonymous signature scheme is that signature schemes are not designed for hiding messages. It is different from a public key encryption scheme. For a secure key-privacy-enabled encryption scheme, an attacker (i.e. the one who wants to find out the identity of the decryptor) has access to both the message and the corresponding ciphertext (and of course the public keys of all decryptors in a system). For constructing a secure anonymous signature scheme, on the other hand, we need to consider the impacts of messages to the anonymity of signatures more carefully. For example, if a signature and the corresponding message are given, it is impossible to have a signature scheme be anonymous because the signature is publicly verifiable and the number of public keys in a system is usually limited. Another example, if the message of a challenge signature is not given but the message space is small, it would still be easy to find out the identity of the signer by searching over all the possible messages for each possible signer. In the following sections, we will see that we tackle the problems related to message characteristics (such as message space and message distribution) from both definitions and techniques. On definitions, we define the exact meaning of an anonymous signature scheme with respect to the message characteristics. On techniques, we will propose some major ones for making sure that message characteristics would not compromise signer anonymity.

Notice that signer anonymity is not the same as *sender anonymity* while the latter is not new. In signcryption schemes with key privacy [9,26], or in designated verifier signature schemes [18,19], the identity of the sender is protected (i.e. sender anonymity) using the intended decryptor/verifier's public key. Their techniques are similar to that of key-privacy-enabled encryption schemes [2]. An anonymous signature scheme, on the other hand, does not have an intended recipient when a signature is generated. It solely focuses on the signer anonymity of a signature scheme itself.

The term, signer anonymity, can also be found in literature related to group signature [11,4] and ring signature [24,12]. But their meaning of signer anonymity is more precisely to be read as 1-out-of-n (or t-out-of-n for threshold settings) signer anonymity, where n is fixed for each given group/ring signature. These schemes have a set of n signers defined by each signature and the signer anonymity of the signature is to prevent anyone from finding out the actual signer out of these n possible signers. In addition, the computational complexity of these schemes is in proportion to the size n of the signer set defined by each of these signatures. An anonymous signature scheme, on the other hand, is rather a conventional signature scheme with an additional property – signer anonymity. The computational complexity of a signature is independent of the number of public keys in a system, and the level of anonymity is independent of the number of public keys in a system either (provided that all the public keys are defined appropriately according to the specification of the anonymous signature scheme). If we extend our notion of signer anonymity to group signature or ring signature, we are actually making the *group* of possible signers anonymous. That is, given a group/ring signature, an attacker cannot find out who is in the signer group or who is not.

3 A Static Security Model for Signer Anonymity

Definition 1. *A digital signature scheme is a tuple of four algorithms denoted by* $(\mathcal{K}, \mathcal{M}, \mathcal{S}, \mathcal{V})$*.*

1. *The key generation algorithm* $\mathcal{K}$ *is a randomized algorithm which on input* 1^k*, where* $k \in \mathbb{N}$ *is a security parameter, returns in polynomial time a pair* (pk, sk) *of matching public and secret keys.*
2. *The message space generator* $\mathcal{M}$ *is an algorithm which on input a public key* pk *returns in polynomial time a set* M *(called the message space with respect to* pk*). Formally, the output is a description of* M *and for simplicity, we denote* M *by* $\mathcal{M}(pk)$*.*
3. *The signing algorithm* $\mathcal{S}$ *is a (possibly randomized) algorithm which on input* 1^k*, a message* m *and the secret key* sk *returns in polynomial time a signature* σ *for* m*.*
4. *The verification algorithm* $\mathcal{V}$ *is a deterministic algorithm which on input* 1^k*, a message* m*, the public key* pk*, and a candidate signature* σ *for* m *returns in polynomial time a bit indicating the validity of the signature.*

(*Correctness.*) We require that $\mathcal{V}(1^k, m, pk, \mathcal{S}(1^k, m, sk)) = 1$ for any $(pk, sk) \leftarrow \mathcal{K}(1^k)$ and $m \in \mathcal{M}(pk)$.

From the definition above, we explicitly specify that the message space is defined by the public key. In the past, this is usually assumed but not explicit and is often considered to have a common message space for all keys in a system. In this paper, we explicitly define the message space as it is important to our discussions of signer anonymity. Another possible definition of $\mathcal{M}$ is to consider it as a randomized algorithm which generates messages directly. In other words, the message distribution is also specified by the scheme. However, it is unnatural. In practice, a signature scheme only has the message space defined with respect to each public key. It is up to the specific application to decide how the messages are to be drawn from the message space. Therefore, we leave the distribution of messages undefined and specify it only when it comes into place for ensuring signer anonymity.

A signature aims to provide message authentication and non-repudiation, so in the literature, most of the results are focusing on the impossibility of producing forgeries. The theme of this paper is to consider an *auxiliary* property for digital signature: signer anonymity. In the following, we specify a basic model which captures our fundamental notion of signer anonymity. For simplicity, we omit the expression of 1^k from the inputs of $\mathcal{S}$ and $\mathcal{V}$ in the rest of the paper.

3.1 Static Model

Definition 2. *Let $\mathcal{SD} = (\mathcal{K}, \mathcal{M}, \mathcal{S}, \mathcal{V})$ be a digital signature scheme. Suppose the key generation algorithm is run twice with the security parameter k, and $(pk_0, sk_0) \leftarrow \mathcal{K}(1^k)$ and $(pk_1, sk_1) \leftarrow \mathcal{K}(1^k)$ are generated. $\mathcal{SD}$ is said to produce computationally indistinguishable signatures (or signatures with signer anonymity in the static model) if for every probabilistic polynomial time (PPT) algorithm $\mathcal{D}$, every positive polynomial $p(\cdot)$, and all sufficiently large k's,*

$$|\Pr[\mathcal{D}(1^k, pk_0, pk_1, \sigma_0) = 1] - \Pr[\mathcal{D}(1^k, pk_0, pk_1, \sigma_1) = 1]| < \frac{1}{p(k)} \qquad (1)$$

where $\sigma_0 \leftarrow \mathcal{S}(m_0, sk_0)$, $\sigma_1 \leftarrow \mathcal{S}(m_1, sk_1)$ and $m_0 \in_R \mathcal{M}(pk_0)$, $m_1 \in_R \mathcal{M}(pk_1)$.

By $x \in_R X$, we mean that an element x is randomly chosen from a set X. A weaker version of Def. 2 is to restrict that the message spaces of both public keys are identical and the signatures σ_0 and σ_1 are signatures of the same message, that is, $m_0 = m_1$. This model looks intuitive but restrictive when compared with the definition above. In many cases, the message spaces for different public keys are not the same or their message distributions are not identical. For example, the basic RSA signature scheme [23] reviewed below may have the size of the message space depend on the value of the RSA modulus.

3.2 Discussions

A *message-recoverable* signature scheme, such as PSS-R [7], allows the message of each of its signatures to be recovered directly from the signature once the corresponding public key is given while having negligible chance to have a message recovered from the signature if an incorrect public key is supplied. In Def. 2, since public keys are known to $\mathcal{D}$, we can see that a *message-recoverable* signature scheme cannot be anonymous.

Although messages m_0 and m_1 are unknown to the distinguisher $\mathcal{D}$, the corresponding message spaces are publicly known (since $\mathcal{M}$, pk_0 and pk_1 are known). Hence for satisfying Def. 2, it is required that all message spaces should be sufficiently large so that it is negligible for $\mathcal{D}$ to guess correctly the message. One may consider that every message space should have at least 2^k messages. We will give a more precise specification to the message space. One should also note that the size of the message space is a necessary requirement to the anonymity of a signature scheme, but it is not sufficient.

On the signature spaces, Def. 2 also indicates that $\mathcal{D}$ should not be able to distinguish computationally a signature from one space to another. As a counterexample, if the signature space correlates to the length of the corresponding public key (mentioned earlier in the introduction section), $\mathcal{D}$ may be able to compromise the anonymity of a signature from this information.

In the next section, we will see some concrete signature schemes and show that they are not signer anonymous under Def. 2. We will then give an even stronger definition of signer anonymity and propose some techniques for transforming those schemes into signer anonymous versions which can be proven under the stronger notion.

4 Signature Signatures that are Not Signer Anonymous

In this section, we review some of the commonly used signature schemes and show that they are in general not signer anonymous.

4.1 The Basic RSA Signature Scheme

In the following, we show that unless intentionally specified, the basic RSA signature scheme [23] (the primitive one without using hash function), in its general use, is not signer anonymous according to Def. 2.

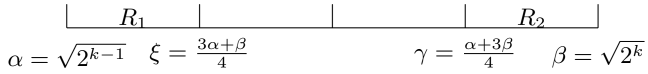
Consider two signers $Signer_0$ and $Signer_1$ with RSA moduli N_0 and N_1, respectively. Without loss of generality, let $N_0 > N_1$. If the two moduli are of different length, it is obvious that signatures generated by the two signers can easily be identified by checking the length of a given signature. Even if N_0 and N_1 are of equal length, we can still distinguish signatures for most of the cases. In the following, we elaborate this in detail.

Let us evaluate the probability that a signature of $Signer_0$ falls into the range of $\mathbb{Z}_{N_0} - \mathbb{Z}_{N_1}$. Let $\Delta = N_0 - N_1$. The probability that a signature of $Signer_0$ falls into $\{N_1, \cdots, N_0 - 1\}$ will be Δ / N_0. This value is upper bounded by $\Delta / 2^{k-1}$ if $|N_0| = k$. Hence if $|\Delta|$ is in the order of $\log(k)$, then the probability will be negligible for sufficiently large k. This is the case when we say that N_0 and N_1 are "very close" to each other. In this case, the basic RSA signature scheme may be anonymous. However, this is true only if all message spaces in the system are *dense* in the corresponding ranges, for example, every element in $\mathbb{Z}_{N_i}$, $i = 0, 1$, is valid/meaningful. On the other hand, if the message space of $Signer_0$ or $Signer_1$ is *sparse* in $\mathbb{Z}_{N_i}$, $i = 0/1$, that is, there are only a few elements in $\mathbb{Z}_{N_i}$ that are valid (or meaningful) messages. Then the scheme cannot be anonymous. For example, suppose a signature $\sigma = m_0^{d_0} \bmod N_0$ is given where d_0 is the private exponent of $Signer_0$, the distinguisher $\mathcal{D}$ can determine if $Signer_1$ is the actual signer by computing $m' = \sigma^{e_1} \bmod N_1$, where e_1 is the public exponent of $Signer_1$ and then determining if m' is in the message space of $Signer_1$. Since the message space of $Signer_1$ in $\mathbb{Z}_{N_1}$ is sparse, it will have a non-negligible chance that m' is not in the message space, which allows $\mathcal{D}$ to find out the actual signer with non-negligible advantage.

All of the above are concerning about special cases. In the general case where N_0 and N_1 are generated by following a *conventional procedure*, that is, each of N_0 and N_1 is a product of two randomly chosen equal-length primes and $|N_0| = |N_1| = k$, the following theorem implies that with at least a constant probability that a RSA signature can be distinguished successfully (i.e. not signer anonymous under Def. 2).

Theorem 1. *If N_0 and N_1 are generated by following the conventional procedure, then the probability that $|N_0 - N_1| \geq 2^{k-2}$ is at least $\frac{1}{400}$.*

Proof. Suppose N_0 and N_1 are generated by following the conventional procedure, that is, randomly generate two equal-length primes p_0 and q_0 and set $N_0 = p_0 q_0$ such that $|N_0| = k$ and do the same for N_1. Without loss of generality, consider $N_0 > N_1$. We show that with at least a constant probability (i.e. non-negligible probability), a signature of $Signer_0$ falls into the range $\{N_1, \cdots, N_0 - 1\}$, and therefore signatures generated by $Signer_0$ and $Signer_1$ are generally distinguishable even if N_0 and N_1 are of the same length.



R_1
R_2
$\alpha = \sqrt{2^{k-1}}$
$\xi = \frac{3\alpha+\beta}{4}$
$\gamma = \frac{\alpha+3\beta}{4}$
$\beta = \sqrt{2^k}$

Fig. 2. The Range of RSA Prime Factors

Consider a RSA modulo $N = pq$ where p and q are random prime of the same length. For $2^{k-1} < N < 2^k$, we have $\alpha = \sqrt{2^{k-1}} < p, q < \beta = \sqrt{2^k}$. The range is illustrated in Fig. 2.

The number of primes that are less than or equal to an integer n is roughly $n/\ln(n)$ where $\ln(\cdot)$ denotes the natural logarithm. This implies that the prime density $(1/\ln(n))$ is in decreasing order. If we randomly choose a prime $\alpha < x < \beta$, the probability that it is in R_1 is greater that $1/4$. Hence, we have

$$\Pr[p \text{ and } q \text{ are in } R_1] > \frac{1}{16}.$$

Using the prime density function we can also calculate for a randomly chosen prime $\alpha < x < \beta$,

$$\begin{aligned} \Pr[x \text{ is in } R_2] &\approx \frac{\frac{\beta}{\ln(\beta)} - \frac{\gamma}{\ln(\gamma)}}{\frac{\beta}{\ln(\beta)} - \frac{\alpha}{\ln(\alpha)}} \\ &\approx \frac{\frac{2^{k/2}}{k/2} - \frac{c_1 2^{k/2}}{k/2+\ln(c_1)/\ln(2)}}{\frac{2^{k/2}}{k/2} - \frac{2^{k/2}}{(k-1)/\sqrt{2}}} \end{aligned}$$

$$\approx \frac{\frac{1}{k} - \frac{c_1}{k+2\ln(c_1)/\ln(2)}}{\frac{1}{k} - \frac{1}{\sqrt{2}(k-1)}}$$

$$\approx \frac{\sqrt{2}(1-c_1)k^3 + O(k^2)}{(\sqrt{2}-1)k^3 + O(k^2)}$$

where $c_1 = \frac{3+1/\sqrt{2}}{4}$. We then get

$$\Pr[x \text{ is in } R_2] \approx \frac{((\sqrt{2}-1)/4)k^3 + O(k^2)}{(\sqrt{2}-1)k^3 + O(k^2)}$$
$$> \frac{1}{5}$$

for sufficiently large k. Therefore,

$$\Pr[p \text{ and } q \text{ are in } R_2] > \frac{1}{25}.$$

And for any N_0 generated by p_0 and q_0 in R_2 and N_1 generated by p_1 and q_1 in R_1,

$$N_0 - N_1 \geq \gamma^2 - \xi^2 = \frac{\beta^2 - \alpha^2}{2} = 2^{k-2}.$$

Thus with at least a constant probability (i.e. non-negligible probability), N_0 and N_1 are "far away" from each other that leads to the result of having signatures of $Signer_0$ fall into the range $\{N_1, \cdots, N_0 - 1\}$ with non-negligible chance. Therefore signatures generated by $Signer_0$ and $Signer_1$ are generally distinguishable even if N_0 and N_1 are of the same length. □

4.2 PSS

Based on the results above, we can see that PSS [7] is not signer anonymous either. Below are the details.

Let $k \in \mathbb{N}$ be a security parameter. There are two additional security parameters k_0 and k_1 satisfying $k_0 + k_1 \leq k-1$. As suggested in [7], we can imagine $k = 1024$, $k_0 = k_1 = 128$. Let $h : \{0,1\}^* \rightarrow \{0,1\}^{k_1}$ and $g : \{0,1\}^{k_1} \rightarrow \{0,1\}^{k-k_1-1}$ be two hash functions[3]. Let g_1 be the function on input $w \in \{0,1\}^{k_1}$ returns the first k_0 bits of $g(w)$, and let g_2 be the function which on input $w \in \{0,1\}^{k_1}$ returns the remaining $k - k_0 - k_1 - 1$ bits of $g(w)$.

The key generation algorithm $\mathcal{K}$ is the same as that of the basic RSA: $(pk, sk) \leftarrow \mathcal{K}(1^k)$ where $pk = (N, e)$ and $sk = (N, d)$ with N being a composite of two randomly generated equal-length primes and $|N| = k$. The message space M^{PSS} can be any subset of $\{0,1\}^*$. The signature generation and verification algorithms are described as follows.

[3] As g is actually an expansion function, we can consider it as a one-way function which is viewed as a random oracle for security analysis.

```
$\mathcal{S}(m, sk)$
   1. $r \overset{R}{\leftarrow} \{0,1\}^{k_0}$; $w \leftarrow h(m\|r)$; $r^* \leftarrow g_1(w) \oplus r$
   2. $y \leftarrow 0\|w\|r^*\|g_2(w)$. The first 0-bit is to guarantee that $y$ is in $\mathbb{Z}_N$.
   3. return $\sigma = y^d \bmod N$
```

```
$\mathcal{V}(m, pk, \sigma)$
   1. $y \leftarrow \sigma^e \bmod N$
   2. Break up $y$ as $b\|w\|r^*\|\gamma$ (That is, let $b$ be the first bit of $y$, $w$ the next
      $k_1$ bits, $r^*$ the next $k_0$ bits, and $\gamma$ the remaining bits.)
   3. $r \leftarrow r^* \oplus g_1(w)$
   4. If ($h(m\|r) = w$ and $g_2(w) = \gamma$ and $b = 0$) then return 1
      Else return 0
```

(Analysis.) We can see that PSS has the same problem as the basic RSA, that is, the actual signer of a signature can be found out simply by examining the length of the signature or evaluating the 'gaps' among the ranges of the signature spaces of different signers. Theorem 1 applies directly to PSS. In the following, we examine an extra feature that PSS has.

This feature allows a specific distinguisher $\mathcal{D}$ (in Def. 2) to distinguish the signatures between two different signers even the RSA moduli of these two signers are "very close" (as defined in Sec. 4.1) to each other. Suppose N_0 and N_1 are both k bits long. We construct a distinguish $\mathcal{D}$ in the following way:

> "On input $(1^k, pk_0, pk_1, \sigma)$, compute $y \leftarrow \sigma^{e_0} \bmod N_0$, and break up y as done in STEP 2 of the verification algorithm $\mathcal{V}$ above. If $g_2(w) = \gamma$, output 1; otherwise, output 0."

In the case $\sigma = \sigma_0$, $\Pr[\mathcal{D}(1^k, pk_0, pk_1, \sigma) = 1] = 1$. In the case $\sigma = \sigma_1$, if g is considered as a random oracle [6], then the probability that $g_2(w) = \gamma$ is negligible in $k - k_0 - k_1 - 1$. Using $k = 1024$, $k_0 = k_1 = 128$, we can see that $\mathcal{D}$'s advantage is overwhelming, and $\mathcal{D}$ is in polynomial time.

4.3 Schnorr Signature Scheme [25]

On input a security parameter 1^k, the key generation algorithm $\mathcal{K}$ returns a public key pk which consists of a set of group parameters $\mathcal{I} = (p, q, g, G, h)$ and an element $y \in G$, and a secret key sk which is a random element $x \in_R \mathbb{Z}_q$, such that $y = g^x \bmod p$. In $\mathcal{I}$, p, q are two large primes chosen randomly such that $q|p$–1, G is a subgroup of $\mathbb{Z}_p^*$ with order q, g is a generator of G so that computing discrete logarithms to the base g is difficult, and $h : \{0,1\}^* \rightarrow \{0, 1, \cdots, 2^k - 1\}$ is a hash function where $2^k < q$.

In the original Schnorr signature scheme, the message space can be arbitrarily specified as any subset of $\{0,1\}^*$. For allowing us to specify the minimum size of the message space that an anonymous Schnorr signature scheme should be in the later part of this paper, we quantify the message space. We define the message space generator $\mathcal{M}$ such that on input pk, which is generated by $\mathcal{K}(1^k)$, $\mathcal{M}(pk)$ outputs the description of a message space $M^{Schnorr}$ such that $|M^{Schnorr}| \geq 2^k$. Below are the signature generation and verification algorithms.

Signing algorithm. On input a message $m \in M^{Schnorr}$ and a secret key x, $\mathcal{S}(m, x)$ is computed as follows:

1. Choose a random $w \in_R \mathbb{Z}_q$ and compute $t = g^w \bmod p$.
2. Compute $r = h(t, m)$.
3. Compute $s = w - xr \bmod q$.

The signature for m is the pair (r, s).

Verification algorithm. To verify a signature (r, s) for message m under public key $(\mathcal{I}, y)$, compute $t = g^s y^r \bmod p$ and output 1 if $r = h(t, m)$, otherwise output 0.

Since signers generate their public key pairs independently, it is pretty likely that different signers have their keys under different sets of group parameters. We can see that the scheme is not signer anonymous as identity information will be leaked from the value of s by applying similar arguments to that in Sec. 4.1. Interestingly, in a special case where all signers are sharing a common set of group parameters, the scheme can actually be shown to provide signer anonymity under the random oracle model [6] without any modification. The proof technique is similar to that for Lemma 2.

ElGamal Signature Scheme [13]. The analysis of ElGamal signature scheme is similar to the above. We skip the details in this paper.

In the next section, we define a stronger notion of signer anonymity for digital signature schemes.

5 An Adaptive Security Model for Signer Anonymity

Def. 2 is static as the distinguisher cannot adaptively acquire additional information about the challenging signature from the environment. In the following, we define a stronger model which allows the distinguisher to adaptively obtain signatures generated by the entity who generates the challenging signature.

Definition 3 (SA-CMA). *Let k be a security parameter. A digital signature scheme $\mathcal{SD}$ is signer anonymous against chosen message attack (SA-CMA) if for all sufficiently large k, no PPT adversary (or distinguisher) $\mathcal{D}$ can win the following game with a probability non-negligibly larger than $\frac{1}{2}$. The game is simulated by a challenger.*

1. *(Key Generation Phase.) The challenger runs $\mathcal{K}(1^k)$ multiple times for generating polynomially many public and secret key pairs. All the public keys are accessible by $\mathcal{D}$.*
2. *(Training Phase.) $\mathcal{D}$ adaptively queries the challenger with a public key pk_i and a message $m \in \mathcal{M}(pk_i)$. The challenger produces $\sigma \leftarrow \mathcal{S}(m, sk_i)$ and replies $\mathcal{D}$ with σ if pk_i is generated in the Key Generation Phase; otherwise, a '$\perp$' is returned indicating that signature generation has failed.*

3. *(Key Selection Phase I.) $\mathcal{D}$ picks two public keys from the public keys generated in the Key Generation Phase. We denote these two key pairs by (pk_0, sk_0) and (pk_1, sk_1).*
4. *(Key Selection Phase II.) The challenger gives all the secret keys to $\mathcal{D}$ except sk_0 and sk_1.*
5. *(Challenge Phase.) The challenger tosses a random coin $\varpi \xleftarrow{R} \{0,1\}$, then randomly picks a message $m \in_R \mathcal{M}(pk_\varpi)$, and returns a challenge signature $\sigma \leftarrow \mathcal{S}(m, sk_\varpi)$ to $\mathcal{D}$.*
6. *(Cracking Phase.) $\mathcal{D}$ can still adaptively make signing queries as in the Training Phase but the associated public key with each query can only be pk_0 or pk_1.*
7. *(Output Phase.) At the end of the game, $\mathcal{D}$ outputs a bit ϖ' and wins if $\varpi' = \varpi$.*

$\mathcal{D}$'s advantage is defined as $\mathbf{Adv}^{\mathsf{sa-cma}} = \Pr[\varpi' = \varpi] - \frac{1}{2}$ and $\Pr[\varpi' = \varpi]$ is the probability that $\mathcal{D}$ wins the game. The probability is taken over the coin tosses of both $\mathcal{D}$ and the challenger, including the coin toss for ϖ.

If a scheme satisfies this definition, we say that the scheme is SA-CMA secure.

As the distinguisher $\mathcal{D}$ of the adaptive model has an additional signing oracle to access, the model is obviously stronger than the static one given in Def. 2. Another seemingly "stronger" definition is to let $\mathcal{D}$ perform the Challenge Phase and the Cracking Phase in the following way:

Definition 4. *...*

5. *The challenger tosses a random coin $\varpi \xleftarrow{R} \{0,1\}$.*
6. *$\mathcal{D}$ can adaptively perform the following queries:*
   (a) *$\mathcal{D}$ performs signing queries as in the Training Phase except that now the allowable public keys are pk_0 and pk_1 only.*
   (b) *$\mathcal{D}$ queries a special oracle called challenging oracle. The challenging oracle randomly picks a message $m \in_R \mathcal{M}(pk_\varpi)$, and returns $\sigma \leftarrow \mathcal{S}(m, sk_\varpi)$ to $\mathcal{D}$.*

...

But the following result shows that Def. 3 and Def. 4 are equivalent.

Theorem 2. *If there exists no PPT algorithm that has a non-negligible advantage in winning the game in Def. 3, then there exists no PPT algorithm that has a non-negligible advantage in winning the game in Def. 4.*

The proof below uses the "hybrid" technique described in [14].

Proof. Before we go to the proof we denote the games in Def. 3 and Def. 4 by Game 1 and Game 2, respectively.

The proof is by contradiction. Suppose there exists a polynomial time adversary $\mathcal{D}$ that wins Game 2 with non-negligible advantage by performing $p(k)$ challenging queries where $p(\cdot)$ denotes a polynomial. We construct another polynomial time adversary $\mathcal{D}'$ that wins Game 1 with non-negligible advantage.

$\mathcal{D}'$ runs $\mathcal{D}$ by performing all the actions that $\mathcal{D}$ has made except the challenging queries. $\mathcal{D}'$ uniformly chooses a number i from $(1, 2, \cdots, p(k))$. For the first $i-1$ challenging queries, $\mathcal{D}'$ answers $\mathcal{D}$ with $\sigma_0 = \mathcal{S}(m, sk_0)$ by querying the signing oracle on pk_0 and a message $m \in \mathcal{M}(pk_0)$. For the i-th query, $\mathcal{D}'$ answers $\mathcal{D}$ with the challenge signature σ it has received in Game 1. For the rest $p(k)-i$ challenging queries, it answers $\mathcal{D}$ with $\sigma_1 = \mathcal{S}(m, sk_1)$ by querying the signing oracle on pk_1 and a message $m \in \mathcal{M}(pk_1)$. $\mathcal{D}'$ outputs what $\mathcal{D}$ outputs.

Now let us assess the success rate of $\mathcal{D}'$. For simplicity, we use the subscript (0 or 1) to indicate the secret key used to generate the signature.

Let $\lambda_0 = \Pr[\mathcal{D}(\sigma_0^1, \sigma_0^2, \cdots, \sigma_0^{p(k)}) = 1]$. And let $\lambda_1 = \Pr[\mathcal{D}(\sigma_1^1, \sigma_1^2, \cdots, \sigma_1^{p(k)}) = 1]$. From the assumption, $\mathcal{D}$'s advantage in Game 2 is defined as $\epsilon = \frac{1}{2}(1-\lambda_0) + \frac{1}{2}\lambda_1 - \frac{1}{2} = \frac{1}{2}(\lambda_1 - \lambda_0)$ which is non-negligible. Then

$$\Pr[\mathcal{D} \text{ outputs } 1|b=1] = \frac{1}{p(k)}\sum_{i=1}^{p(k)}(\Pr[\mathcal{D}(\sigma_0^1, \sigma_0^2, ..., \sigma_0^{i-1}, \sigma_1^i, \sigma_1^{i+1}, ..., \sigma_1^{p(k)}) = 1])$$

and

$$\Pr[\mathcal{D} \text{ outputs } 1|b=0] = \frac{1}{p(k)}\sum_{i=1}^{p(k)}(\Pr[\mathcal{D}(\sigma_0^1, \sigma_0^2, ..., \sigma_0^{i-1}, \sigma_0^i, \sigma_1^{i+1}, ..., \sigma_1^{p(k)}) = 1])$$

Finally,

$$\begin{aligned}
\Pr[\mathcal{D}' \text{ wins the game }] &= \Pr[\mathcal{D} \text{ outputs } 0|b=0]\Pr[b=0] + \\
&\quad \Pr[\mathcal{D} \text{ outputs } 1|b=1]\Pr[b=1] \\
&= \frac{1}{2} + \frac{1}{2}(\Pr[\mathcal{D} \text{ outputs } 1|b=1] - \Pr[\mathcal{D} \text{ outputs } 1|b=0]) \\
&= \frac{1}{2} + \frac{1}{2p(k)}(\sum_{i=1}^{p(k)}(\Pr[\mathcal{D}(\sigma_0^1, \sigma_0^2, ..., \sigma_0^{i-1}, \sigma_1^i, \sigma_1^{i+1}, ..., \sigma_1^{p(k)}) = 1]) \\
&- \sum_{i=1}^{p(k)}(\Pr[\mathcal{D}(\sigma_0^1, \sigma_0^2, ..., \sigma_0^{i-1}, \sigma_0^i, \sigma_1^{i+1}, ..., \sigma_1^{p(k)}) = 1])) \\
&= \frac{1}{2} + \frac{\epsilon}{p(k)}.
\end{aligned}$$

So $\mathcal{D}'$ wins the game with a non-negligible advantage and $\mathcal{D}'$ runs in polynomial time. □

The theorem above indicates that once the distinguisher is given access to a signing oracle, then giving it one challenge is equivalent to giving it polynomially many challenges.

6 Modified Signature Schemes for Signer Anonymity

In this section, we propose some modifications on the schemes described in Sec. 4 and show that they are signer anonymous under the adaptive model (i.e. SA-CMA in Def. 3). We start with Schnorr signature scheme and provide the full proof for its signer anonymity. Then we modify the basic RSA signature scheme and subsequently the PSS.

6.1 Extended Schnorr Signature Scheme for Signer Anonymity

The key generation algorithm $\mathcal{K}$ and the message space generator $\mathcal{M}$ are almost the same as the original Schnorr signature scheme described in Sec. 4.3, except that the public key now also contains an additional parameter denoted by $b \in \mathbb{N}$. Let q_{min} and q_{max} denote the lower bound and upper bound of the group orders of all signers, respectively. Let 2^b be an integer which is ℓ bits longer than q_{max} and $\ell = k + 1$. One may imagine $k = 160$ and hence $\ell = 161$. Let $h : \{0,1\}^* \rightarrow \{0, 1, \cdots, 2^k - 1\}$ be a hash function where $2^k < q_{min}$.

For a signer with public key $pk = (\mathcal{I}, b, y)$ and secret key x generated by $\mathcal{K}(1^k)$ where $\mathcal{I} = (p, q, g, G, h)$ and $y = g^x \bmod p$, the signature generation and verification algorithms are as follows. Let n be the largest integer such that $nq < 2^b$.

Signing algorithm. On input a message $m \in \mathcal{M}(pk)$ and secret key x, $\mathcal{S}(m, x)$ is computed as follows:

1. Choose a random $w \in \mathbb{Z}_q$ and compute $t = g^w \bmod p$.
2. Compute $r = h(t, m)$ and then $s = w - xr \bmod q$.
3. Choose a number $\lambda \overset{R}{\leftarrow} \{0, 1, \cdots, n-1\}$ and compute $s' = s + \lambda q$

The signature for m is the pair (r, s').

Verification algorithm. To verify signature (r, s') for message m and public key $(\mathcal{I}, y)$, compute $s = s' \bmod q$ and $t = g^s y^r \bmod p$, and output 1 if $r = h(t, m)$, otherwise, output 0.

Consider two arbitrary signers $Signer_i$ and $Signer_j$ whose sets of group parameters are denoted by $\mathcal{I}_i = (p_i, q_i, g_i, G_i, h)$ and $\mathcal{I}_j = (p_j, q_j, g_j, G_j, h)$, respectively. Let n_i and n_j be the largest integers such that $n_i q_i < 2^b$ and $n_j q_j < 2^b$, respectively. Without loss of generality, we assume $n_i q_i < n_j q_j$.

Lemma 1. *For the extended Schnorr signature scheme above, if signer $Signer_i$ generates a signature (r_i, s'_i) and signer $Signer_j$ generates a signature (r_j, s'_j), then the probability that s'_j is in $\Delta = \{n_i q_i, \cdots, n_j q_j - 1\}$ is at most 2^{-k}.*

Proof. First, note that s'_i and s'_j are uniformly distributed on $\{0, 1, \cdots, n_i q_i - 1\}$ and $\{0, 1, \cdots, n_j q_j - 1\}$, respectively. Second, since $n_j q_j < 2^b$ and $n_i q_i \geq 2^b - q_i$, $n_j q_j - n_i q_i < 2^b - (2^b - q_i) = q_i \leq q_{max}$. Hence,

$$\Pr[s'_j \in \Delta] < q_{max}/(2^b - q_{max}) < 1/2^{l-1} = 1/2^k.$$

□

In the following, we assume that h behaves like a random oracle [6]. If an algorithm $\mathcal{A}$ runs in time at most t and completes successfully with probability at least $\epsilon > 0$, then $\mathcal{A}$ is said to be a (t, ϵ)-algorithm. The probability is taken over the input domain and the coin tosses of $\mathcal{A}$.

Lemma 2. *In the extended Schnorr signature scheme above, suppose for any pair of signers Signer$_i$ and Signer$_j$, $q_i = q_j$. Then if there exists a $(t, \epsilon + \frac{1}{2})$-algorithm (distinguisher) $\mathcal{D}$ which wins the game of Def. 3 after performing at most q_H hash queries and q_S signing queries, there exists a (t', ϵ')-algorithm $\mathcal{F}$ which existentially forges under the chosen message attack [15] a signature after performing at most $q_H + q_S$ hash queries and q_S signing queries, where $t' \leq t + q_K c$ and $\epsilon' \geq (1 - \frac{q_H + q_S}{2^k})(1 - \frac{q_S}{2^k})\frac{\epsilon}{q_K}$ for q_K being some polynomial in k and c being the time required for generating one key pair in the extended Schnorr signature scheme.*

Proof. We construct an algorithm $\mathcal{F}$ which runs $\mathcal{D}$ under a simulated environment of Def. 3 and forges a Schnorr signature.

At the beginning of the simulation, $\mathcal{F}$ is given a security parameter k, a set of group parameters $\mathcal{I} = (p, q, g, G, h)$, a challenge element $\mathsf{y} \in G$, an auxiliary parameter $b \in \mathbb{N}$ and a message space $M^{Schnorr}$ such that $|M^{Schnorr}| \geq 2^k$. $\mathcal{F}$ is to forge a signature $\sigma^* = (r^*, s^*)$ with message $m^* \in M^{Schnorr}$ such that $r^* = h(g^{s^*}\mathsf{y}^{r^*} \bmod p,\ m^*)$ where h is provided as a random oracle by the unforgeability game simulator of $\mathcal{F}$. Note that $\mathcal{F}$ has access to the random oracle of h and a signing oracle corresponding to the challenge public key y. The signing oracle, on input a message $m \in M^{Schnorr}$, returns a signature $\sigma = (r, s)$ such that $r = h(g^s\mathsf{y}^r \bmod p,\ m)$. We denote the random oracle for h by $\mathcal{HO}$ and the signing oracle by $\mathcal{SO}$.

In the Key Generation Phase of the game defined in Def. 3, $\mathcal{F}$ randomly generates q_K–1 public key pairs where q_K is some polynomial in k. For each of the public key pairs, say the i-th, the set of group parameters $\mathcal{I}_i = (p_i, q_i, g_i, G_i, h)$ is generated such that $q_i = q$, $q_i | p_i - 1$, and g_i is the generator of G_i whose order is q_i. Also an element y_i is generated as $g_i^{x_i} \bmod p_i$ where x_i is randomly chosen from $\mathbb{Z}_{q_i}$. The public key of i-th public key pair is set to $pk_i = (\mathcal{I}_i, b, y_i)$ and the corresponding secret key is x_i. Let $\mathcal{L} = \{pk_i\}_{1 \leq i \leq q_K}$ be the set of public keys generated in this phase except pk_j, which instead is assigned to $(\mathcal{I}, b, \mathsf{y})$. The value of j is chosen randomly from 1 to q_K.

In the Training Phase and the Cracking Phase, $\mathcal{F}$ answers all oracle queries made by $\mathcal{D}$. For a hash query, the query is relayed by $\mathcal{F}$ to $\mathcal{HO}$ for an answer. The answer is then relayed back to $\mathcal{D}$. $\mathcal{F}$ also maintains a list Ψ of queried values and their returns. For a signature query with message m in the corresponding message space, there are two cases. Case 1: if the public key is not y, $\mathcal{F}$ follows the signing algorithm of the scheme to generate a signature. This can be done as $\mathcal{F}$ knows the corresponding signing key (or secret key). Case 2: if the public key is y, $\mathcal{F}$ relays the query to $\mathcal{SO}$ and relays the signature back to $\mathcal{D}$. Note that the list Ψ should also be updated for hash values. In addition to these steps, in the Cracking Phase, we will see shortly that $\mathcal{F}$ needs to carry out a few more checkings when relaying queries and answers between $\mathcal{D}$ and the oracles $\mathcal{HO}$, $\mathcal{SO}$ to and fro.

In the Key Selection Phase I, if $\mathcal{D}$ picks two public keys such that none of the keys is y, $\mathcal{F}$ fails and halts. Let the two public keys be $(\hat{\mathcal{I}}_0, b, \hat{y}_0)$, $(\hat{\mathcal{I}}_1, b, \hat{y}_1)$. Suppose $\mathcal{F}$ does not fail and proceeds successfully to the Challenge Phase, $\mathcal{F}$

sets the challenge signature $\sigma^* = (r^*, s^*)$ by randomly picks $r^* \overset{R}{\leftarrow} \{0,1\}^k$ and $s^* \overset{R}{\leftarrow} \{0, 1, \cdots, nq-1\}$ where n is the largest integer so that $nq < 2^b$. If r^* is already in the list Ψ as a queried hash oracle answer, $\mathcal{F}$ fails and halts (we will see below that this event is called $\mathbf{E}_2$). Otherwise, an entry $(\top, r^*)$ is added into the list Ψ, where $\top$ represents some hash input whose value is not known yet but its hash value has been given as r^*.

The simulation proceeds until $\mathcal{D}$ reaches the Output Phase. When $\mathcal{D}$ outputs and halts, $\mathcal{F}$ also halts and outputs nothing. That means $\mathcal{F}$ has failed to forge a signature. However during the Cracking Phase, whenever $\mathcal{D}$ makes a hash query, $\mathcal{F}$ checks if the answer of $\mathcal{HO}$ is r^*. If this is the case and at the same time the hash evaluation is of the form $h(g^{s^*}\mathsf{y}^{r^*} \bmod p,\ m^*)$ where $m^* \in M^{Schnorr}$ and m^* is not involved in a signing query in the Training phase, $\mathcal{F}$ outputs the forged signature $\sigma^* = (r^*, s^*)$ and message m^*, and halts. In addition, during the Cracking Phase, whenever $\mathcal{D}$ makes a signing query with some message $m^* \in M^{Schnorr}$ under y, $\mathcal{F}$ first queries $\mathcal{HO}$ for the value of $h(g^{s^*}\mathsf{y}^{r^*} \bmod p,\ m^*)$. If the hash value is equal to r^* and m^* is not involved in a signing query in the Training Phase, $\mathcal{F}$ outputs the forged signature $\sigma^* = (r^*, s^*)$ and message m^*, and halts; if the hash value is not r^*, $\mathcal{F}$ then relays the query to $\mathcal{SO}$ and continues the simulation as described above. Note that if m^* turns out to have been queried in some signing query during the Training Phase, $\mathcal{F}$ fails and halts (we will see below that this event is called $\mathbf{E}_3$).

Analysis. First of all, it is easy to see that the running time of $\mathcal{F}$ is in polynomial of that of $\mathcal{D}$ and $\mathcal{F}$ perfectly simulates the game of Def. 3 except during the Challenge Phase. In this phase, the challenger in a real game (that is, $\mathcal{F}$ in the simulated game described above) should have randomly picked a key among two given public keys, then picked a message randomly from the message space corresponding to the chosen public key and generated a challenge signature accordingly. In the following, we show that it is indistinguishable from $\mathcal{D}$'s point of view between the Challenge Phase of a real game and that of the simulated game by $\mathcal{F}$. Essentially, we show that in the simulated game, given a pair (r^*, s^*), it is equally likely to have a message $m \in M^{Schnorr}$ which produces a valid signature equal to (r^*, s^*) no matter which of the public keys $(\hat{\mathcal{I}}_0, b, \hat{y}_0)$ and $(\hat{\mathcal{I}}_1, b, \hat{y}_1)$ is corresponded with.

To show this, we investigate the distribution of the messages which produce a signature (r^*, s^*) with respect to each of $(\hat{\mathcal{I}}_0, b, \hat{y}_0)$ and $(\hat{\mathcal{I}}_1, b, \hat{y}_1)$. For each of $(\hat{\mathcal{I}}_{\varpi^*}, b, \hat{y}_{\varpi^*})$, $\varpi^* = 0, 1$, define a distribution

$$M_{\varpi^*} = \{m\ :\ r^* \leftarrow h(g^{s^*}\hat{y}_{\varpi^*}^{r^*} \bmod p,\ m),\ m \overset{R}{\leftarrow} \{0,1\}^\ell\}.$$

Under the assumption that h is a random function [6], both distributions M_0 and M_1 are uniform, and both M_0 and M_1 have the same expected number of messages which is equal to $|M^{Schnorr}|/2^k$. From the fact that $\log_2(|M^{Schnorr}|) \geq k$, we have at least half chance (derived from $1 - (1 - 2^{-k})^{|M^{Schnorr}|} \geq 1/2$) that the challenge signature $\sigma^* = (r^*, s^*)$, generated by $\mathcal{F}$ in the Challenge Phase of the simulated game above, is a valid signature of some message. Furthermore, as h behaves likes a random oracle with the range of 2^k possible values, it is negligible for $\mathcal{D}$ to find out if the challenge signature is valid or not. Hence from $\mathcal{D}$'s point of view, when given a pair (r^*, s^*), it is equally likely to be a valid signature no matter it is generated by a key corresponding to $(\hat{\mathcal{I}}_0, b, \hat{y}_0)$ or $(\hat{\mathcal{I}}_1, b, \hat{y}_1)$.

Let $\mathbf{E}_1$ be the event that the hash evaluation

$$r^* \leftarrow h(g^{s^*} \hat{y}_{\varpi^*}^{r^*} \bmod p,\ m^*) \tag{2}$$

is carried out during the cracking phase where $\varpi^* = 0/1$. If event $\mathbf{E}_1$ does not occur, by the random oracle assumption, it has not been decided (by the game simulator $\mathcal{F}$) on which message m^* will make Eq. (2) hold. Hence $\mathcal{D}$ has no advantage in winning the game. If $\mathbf{E}_1$ occurs and $\hat{y}_{\varpi^*} = \mathsf{y}$, then $\mathcal{F}$ wins the game of existential unforgeability against chosen message attack.

Since the position of $(\mathcal{I}, b, \mathsf{y})$ in $\mathcal{L}$ is randomly chosen, the probability of selecting $(\mathcal{I}, b, \mathsf{y})$ in Key Selection Phase I is $2/q_K$. Due to the same reason, in event $\mathbf{E}_1$, the chance that $\hat{y}_{\varpi^*} = \mathsf{y}$ is $1/2$. Note that $\Pr[\mathcal{D} \text{ wins}] \geq \epsilon + 1/2$. Let $\Pr[\mathcal{D} \text{ wins } |\mathbf{E}_1] = \lambda + 1/2$. We have

$$\begin{aligned} \epsilon + \frac{1}{2} &\leq \Pr[\mathcal{D} \text{ wins}] \\ &= (\lambda + \frac{1}{2})\Pr[\mathbf{E}_1] + \Pr[\mathcal{D} \text{ wins } |\overline{\mathbf{E}_1}]\Pr[\overline{\mathbf{E}_1}] \\ &= (\lambda + \frac{1}{2})\Pr[\mathbf{E}_1] + \frac{1}{2}\Pr[\overline{\mathbf{E}_1}]. \end{aligned}$$

Hence $\lambda\Pr[\mathbf{E}_1] \geq \epsilon$. Since $\epsilon > 0$, we have $0 < \lambda \leq 1/2$. Therefore $\Pr[\mathbf{E}_1] \geq 2\epsilon$.

To find out the lower bound of the winning probability of $\mathcal{F}$, we only have two events left to evaluate, that is, the chance that $\mathcal{F}$ fails due to the following two events.

Event $\mathbf{E}_2$: During the Challenge Phase, r^* is found to be in the list of Ψ.

Event $\mathbf{E}_3$: During the Cracking Phase, if evaluation $r^* \leftarrow h(g^{s^*} \mathsf{y}^{r^*} \bmod p, m^*)$ occurs while m^* has been involved in a signing query during the Training Phase.

Since r^* is randomly chosen from $\{0,1\}^k$ and h is a random function, we have $\Pr[\mathbf{E}_2] \leq \frac{q_H + q_S}{2^k}$. Similarly, we have $\Pr[\mathbf{E}_3] \leq \frac{q_S}{2^k}$.

Combining all the events above, they include the case that y is one of $\hat{y}_0$ and $\hat{y}_1$, the case that (r^*, s^*) is a valid signature of y, $\mathbf{E}_1$ occurs, the case that y is involved in the event $\mathbf{E}_1$, the case that r^* is not in the list Ψ during the Challenge Phase (i.e. $\overline{\mathbf{E}_2}$), and the case that the forged message m^* has not been involved in any signing query during the Training Phase (i.e. $\overline{\mathbf{E}_3}$), we have

$$\Pr[\mathcal{F} \text{ wins}] \geq (1 - \frac{q_H + q_S}{2^k})(1 - \frac{q_S}{2^k})\frac{\epsilon}{q_K}.$$

On the running time of $\mathcal{F}$, we can see that besides running $\mathcal{D}$, $\mathcal{F}$ needs to generates $q_K - 1$ key pairs during the Key Generation Phase and at most q_S

additional hash queries during the Cracking Phase. Let c be the time required for generating one key pair. The running time of $\mathcal{F}$ is at most $t + q_K c$. Also $\mathcal{F}$ performs at most $q_H + q_S$ hash queries and q_S signing queries. □

Theorem 3. *The extended Schnorr signature scheme described above is SA-CMA secure.*

Proof. Without loss of generality, suppose in the game of Def. 3, the distinguisher $\mathcal{D}$ picks the public keys corresponding $Signer_i$ and $Signer_j$ in the Key Selection Phase I, and $Signer_j$ is picked by the challenger in the Challenge Phase. We follow the notations used above and in the proof of Lemma 1, we assume that $n_i q_i < n_j q_j$. Let $\mathbf{E}$ be the event that $s'_j \notin \Delta$. In other words, $\mathbf{E}$ is the event that $s'_j \in \{0, 1, \cdots, n_i q_i - 1\}$, that is, in the same domain as $Signer_i$ has been picked by the challenger. According to Lemma 2, we have $\Pr[\mathcal{D}$ wins the game $|\mathbf{E}] \leq \frac{1}{2} + \epsilon(k)$ under the assumption that the extended Schnorr signature scheme is existentially unforgeable [15], where ϵ is a negligible function. Since $\Pr[\mathbf{E}] \leq 1$, we have

$$\Pr[\mathcal{D} \text{ wins the game } \wedge \mathbf{E}] \leq \frac{1}{2} + \epsilon(k) \qquad (3)$$

According to Lemma 1, we have $\Pr[\overline{\mathbf{E}}] \leq 2^{-k}$. Since $\Pr[\mathcal{D}$ wins the game $|\overline{\mathbf{E}}] \leq 1$, we have

$$\Pr[\mathcal{D} \text{ wins the game } \wedge \overline{\mathbf{E}}] \leq 2^{-k} \qquad (4)$$

Combining Eq. (3) and (4), we have

$$\Pr[\mathcal{D} \text{ wins the game }] \leq \frac{1}{2} + \epsilon(k) + 2^{-k}$$

□

The extended Schnorr signature scheme still maintains existential unforgeability against adaptive chosen message attack (euf-cma) [15], namely, given a signing oracle, an adversary cannot forge a signature for a message m which has not been queried to the signing oracle before. However, the extended scheme does not satisfy the strong unforgeability [5,1], namely, given a signing oracle, an adversary cannot forge a valid pair of message m and signature σ which has not been a query output of the signing oracle for m before.

6.2 Extended RSA Signature Scheme for Signer Anonymity

In our extended RSA signature scheme, we set up a common message space and signature space for all signers regardless of the values of their RSA moduli. The technique is borrowed from Rivest, Shamir and Tauman in the context of a ring signature scheme [24]. It expands the message spaces of all signers to a common domain $\{0, 1\}^b$ such that 2^b is significantly greater than the RSA moduli (denoted by N_i, $i = 1, 2, \cdots$) of all signers (e.g. b is 160 bits longer than the largest N_i of all signers). An extended trapdoor one-way permutation g_i over $\{0, 1\}^b$ with respect to the signer i's RSA modulus N_i is defined as follows. For any b-bit message m, define nonnegative integers q_i and r_i so that $m = q_i N_i + r_i$ and $0 \leq r_i < N_i$. Then

$$g_i(m) = \begin{cases} q_i N_i + (r_i^{d_i} \bmod N_i) & \text{if } (q_i+1)N_i \le 2^b \\ m & \text{else.} \end{cases}$$

where d_i is the RSA private exponent. The function g_i is a one-way trapdoor permutation over $\{0,1\}^b$ and its inverse function g_i^{-1} is defined as

$$g_i^{-1}(m) = \begin{cases} q_i N_i + (r_i^{1/d_i} \bmod N_i) & \text{if } (q_i+1)N_i \le 2^b \\ m & \text{else.} \end{cases}$$

We then apply the traditional hash-then-sign strategy by using a hash function to map messages from the signer's message space to the common domain $\{0,1\}^b$. Note that the extended RSA signature is euf-cma [15] under the random oracle model. On signer anonymity, we have the following theorem.

Theorem 4. *Let k be a system parameter. Suppose the extended RSA scheme described above has the common domain $\{0,1\}^b$ such that $b - |N_{max}| \ge k$, where N_{max} is the largest value of RSA moduli of all signers. If the message space of each of the signers has at least 2^k messages, then the scheme is SA-CMA secure (with respect to Def. 3).*

Proof (Sketch). Let h be a random oracle (hash function) that maps messages from the signer's message space to the common domain $\{0,1\}^b$. As h is a random oracle, for any signer with RSA modulus N_i and for any message m from the signer's message space, the probability that $g_i(h(m)) = h(m)$ is at most 2^{-k}. The proofing techniques of Lemma 1, Lemma 2 and Theorem 3 can then be used subsequently.

First, if a signer $Signer_i$, whose RSA modulus is N_i, and signer $Signer_j$, whose RSA modulus is N_j, generate message-signature pairs $(m_i, g_i(h(m_i))$ and $(m_j, g_j(h(m_j)))$, respectively, then the probability that $g_i(h(m_i))$ falls in the range $[2^b - N_j + 1, 2^b - N_i]$ is negligible (wlog, assuming that $N_i < N_j$). This can be shown using the proofing technique for Lemma 1.

Second, for the two signers above, if a signature falls in the range $[1, 2^b - N_j]$, by applying the proofing technique for Lemma 2, we can show that $\mathcal{D}$ of the game in Def. 3 will have negligible chance of distinguishing signatures between the two signers as the extended RSA scheme is euf-cma [15].

Finally, by employing the combining technique of the proof of Theorem 3, we conclude that the extended RSA scheme above is SA-CMA secure. □

6.3 Extended PSS for Signer Anonymity

There are two phases. In the first phase, the domain expansion technique used in the extended RSA signature scheme is employed. In the second phase, we solve the problem brought in by the extra feature of PSS discussed in Sec 4.2. Our solution is to conceal the special format of y. Details are as follows.

Let g_i be the extended trapdoor one-way permutation defined above in Sec. 6.2. It corresponds to the public key of signer i. Replace the original hash function $g : \{0,1\}^{k_1} \to \{0,1\}^{k-k_1-1}$ with another hash function $\rho : \{0,1\}^{k_1} \to$

$\{0,1\}^{b-k_1}$. Let ρ_1 be a function which on input $w \in \{0,1\}^{k_1}$ returns the first k_0 bits of $\rho(w)$, and let ρ_2 be a function which on input $w \in \{0,1\}^{k_1}$ returns the remaining $b - k_1 - k_0$ bits of $\rho(w)$. Let $\zeta : \{0,1\}^* \rightarrow \{0,1\}^{b-k_1-k_0}$ be another hash function. We then follow the other notations used in Sec. 4.2. The message space $M^{PSS} \subseteq \{0,1\}^*$ is assumed to contain at least 2^k messages. For security analysis, all hash functions are assumed to behave like random oracles.

The key generation algorithm is the same as the original PSS scheme except that we now have an additional parameter b. Below are the signature generation and verification algorithms which maintain the unforgeability of the scheme in terms of euf-cma [15] under the random oracle model.

```
$\mathcal{S}(m, sk_i)$
    1. $r \overset{R}{\leftarrow} \{0,1\}^{k_0}$; $w \leftarrow h(m\|r)$; $r^* \leftarrow \rho_1(w) \oplus r$
    2. $y \leftarrow w\|r^*\|\rho_2(w) \oplus \zeta(m)$
    3. return $\sigma = g_i(y)$
```

```
$\mathcal{V}(m, pk_i, \sigma)$
    1. $y \leftarrow g_i^{-1}(\sigma)$
    2. Break up $y$ as $w\|r^*\|\gamma$ (That is, let $w$ be the first $k_1$ bits, $r^*$ the next $k_0$
       bits, and $\gamma$ the remaining bits.)
    3. $r \leftarrow r^* \oplus \rho_1(w)$
    4. If ($h(m\|r) = w$ and $\rho_2(w) \oplus \zeta(m) = \gamma$) then return 1
       Else return 0
```

Theorem 5. *Let k be a system parameter. Suppose the extended PSS scheme described above has the common domain $\{0,1\}^b$ such that $b - |N_{max}| \geq k$, where N_{max} is the largest value of RSA moduli of all signers. If the message space of each of the signers has at least 2^k messages, then the scheme is SA-CMA secure.*

Proof (Sketch). In case h, ρ and ζ are random functions (i.e. behaving like random oracles), y is random in $\{0,1\}^b$. Then similar techniques to those used in the proof sketch of Theorem 4 above can be used to prove this theorem in a straightforward way. □

7 Applications

In the introduction of this paper, we have described some applications of anonymous signature schemes. In this section, we provide more details on how to use anonymous signature to enhance user privacy for key exchange. We also propose a new anonymous paper review system which uses anonymous signature to enhance the anonymity of the paper review process against collusion between organizers and reviewers.

7.1 Anonymous Key Exchange

As shown in Fig. 1 and discussed in the introduction section, the protocol cannot provide client anonymity if the Different Domain Attack is feasible. In order to make it client anonymous, we modify the last message flow from A to B by using an anonymous signature scheme and change the message to

$$A \rightarrow B \ : \ Sig_A(h(ID_B, count, \sigma, r_B))$$

where $h : \{0,1\}^* \rightarrow \{0,1\}^k$ is a hash function which behaves like a random oracle.

The example above is an anonymous key transport protocol. Next, we construct an anonymous key *exchange* protocol which not only ensures the anonymity of the client but also allows the client and the server to establish a session key from both of their session key contributions. The protocol is based on a key exchange protocol called "SIG-DH" [10] which is a signature-based variation of the Diffie-Hellman key exchange protocol with provable security against various active attacks defined in the Canetti-Krawczyk model [10].

Anonymous SIG-DH Protocol: (Fig. 3)
Let k be a security parameter. Let G be a group generated by g with large prime order q so that computing discrete logarithms to be base g is difficult. Let $H : \{0,1\}^* \rightarrow \{0,1\}^{3k}$ be a hash function. Each party has a secret signing key for a signature algorithm Sig. By $Sig_A(m)$, we mean the signature on message m generated by party A with identity $ID_A \in \{0,1\}^k$. Assume the public keys of all parties in the system are publicly known. Let E be a block cipher (e.g. AES [22]) of block size k. Suppose a client (the initiator) A and a server (the responder) B already have a session-id s shared. We will explain shortly on how the session-id s is established. The following protocol is carried out between them.

1. A randomly chooses a temporal identity $alias \in_R \{0,1\}^k$, $x \in_R \mathbb{Z}_q$, and sends $(alias, s, \alpha = g^x)$ to B.
2. Upon receipt of $(alias, s, \alpha)$, B randomly chooses $y \in_R \mathbb{Z}_q$, then computes $\kappa_1 \| \kappa_2 \| \kappa_3 \leftarrow H(\alpha^y)$ such that $|\kappa_i| = k$ for $i = 1, 2, 3$, erases y, and sends to A the message $(B, s, \beta = g^y)$ together with $SIG_B(B, s, \beta, \alpha, alias)$.
3. Upon receipt of $(B, s, \beta = g^y)$ and B's signature, A computes $\kappa_1' \| \kappa_2' \| \kappa_3' \leftarrow H(\beta^x)$, erases x, and verifies the signature. If the signature is valid, A sends to B the message $(alias, s, C_1 = E_{\kappa_1}(A))$ together with its signature $\sigma = Sig_A(h(alias, A, s, \alpha, \beta, B, \kappa_2'))$ where $h : \{0,1\}^* \rightarrow \{0,1\}^k$ is a hash function. A outputs the session key κ_3' under session-id s.
4. Upon receipt of $(alias, s, C_1)$ and a signature σ, B computes $A' = E_{\kappa_1}^{-1}(C_1)$, and verifies the identity A' (e.g. for access control) and signature σ. If all verifications are passed, B outputs the session key κ_3 under session-id s.

(*Analysis.*) The protocol described above (Fig. 3) supports anonymity of the client A if Sig is an anonymous signature scheme. In the protocol, all hash functions are assumed to behave like random oracles. The session-id s should also be randomly selected each time for ensuring A's anonymity. As suggested

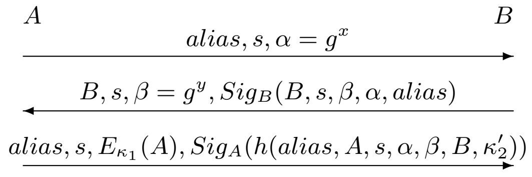

A
B
$alias, s, \alpha = g^x$
$B, s, \beta = g^y, Sig_B(B, s, \beta, \alpha, alias)$
$alias, s, E_{\kappa_1}(A), Sig_A(h(alias, A, s, \alpha, \beta, B, \kappa_2'))$

Fig. 3. Anonymous SIG-DH Protocol

by the authors of [10], in practice, the session-id s can be a pair (s_1, s_2) where s_1 is a value randomly chosen by A such that it is different from the values in other of A's sessions and s_2 is randomly chosen by B in a similar way. These values can be exchanged by the parties as a prologue [16]. Alternatively, s_1 can be included by A in the first message of the protocol, and s_2 be included by B in the second message.

The protocol assumes that the signature verification keys of all parties are publicly known. In practice, we can add the client's certificate into the encryption in the third message provided that the certificates of all clients are of the same length. Also, we assume that the server does not know the client at the beginning of the communication. In case it is already known, the encryption operation in the third message can be removed from the protocol.

Comparing with the original "SIG-DH" protocol [10], the anonymous version proposed above has an additional message component κ_2 in the signature of A. κ_2 is used for satisfying the anonymity requirement of an anonymous signature scheme, that is, preventing an adversary from compromising A's anonymity by searching through the list of all possible 'messages' of the signature.

The two anonymous key transport/exchange protocols described above can be used by a mobile device to communicate with a base station anonymously without being tracked or identified by any eavesdroppers, other mobile devices or foreign base stations.

7.2 Anonymous Paper Review

The process of a current anonymous paper review system is to have authors submit their papers and authorship information to a conference organizer such that their papers should be fully anonymous, with no author names, affiliations, acknowledgements, or obvious references. The organizer then keeps the authorship information of each paper secret from the reviewers and only disseminates those anonymous papers to reviewers for review. However, the system has no protection against collusion between the organizer and a reviewer. The organizer or some insiders in the organizing institute, for example a graduate student who is responsible for maintaining the paper submission server, may leak the authorship information of some papers to the reviewer. In the following, we describe a method of using anonymous signature to solve the collusion problem.

Consider the paper submission server is now a bulletin board which posts and timestamps any message received. Once posted, the message cannot be altered.

Let $Paper_A$ be a paper which is fully anonymous. Let A be the identity of the paper's author and assume that each author already has his public key (for signature verification) published. To submit the paper $Paper_A$, the author randomly picks a long binary string $r \in \{0,1\}^k$ where k is the security parameter, and generates a signature $\sigma_A = \mathsf{AnonSig}_A(h(Paper_A, r))$ using his anonymous signature generation algorithm denoted by $\mathsf{AnonSig}_A$ on the message $h(Paper_A, r)$ where $h : \{0,1\}^* \rightarrow \{0,1\}^k$ is a hash function which behaves like a random oracle. The author posts $Paper_A$ and σ_A onto the bulletin board for review. When all the reviews are completed and the acceptance decision on each paper has been made, the decision will be posted on the bulletin board. If $Paper_A$ is accepted, the author A will reveal the value of r for claiming his authorship on $Paper_A$. From this point on, everyone is able to verify his authorship using σ_A, $(Paper_A, r)$ and A's public key.

Analysis: During the review stage, no author has given out any authorship information and the secrecy of r prevents anyone from identifying the signer of σ_A. This new system can even make those just submitted papers and their signatures public. In this way, it will help authors claim to be the first authors of some new results without compromising the process of anonymous review, as their papers are timestamped when they are first submitted for review. In addition, it will also help discover parallel submissions.

References

1. J.H. An, Y. Dodis, and T. Rabin. On the security of joint signature and encryption. In *Proc. EUROCRYPT 2002*, pages 83–107. Springer-Verlag, 2002. LNCS 2332.
2. M. Bellare, A. Boldyreva, A. Desai, and D. Pointcheval. Key-privacy in public-key encryption. In *Proc. ASIACRYPT 2001*, pages 566–582. Springer-Verlag, 2001. LNCS 2248.
3. M. Bellare, J. Garay, R. Hauser, A. Herzberg, H. Krawczyk, M. Steiner, G. Tsudik, and M. Waidner. iKP – A family of secure electronic payment protocols. In *Proc. of the First USENIX Workshop on Electronic Commerce*, pages 89–106, New York, 1995.
4. M. Bellare, D. Micciancio, and B. Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In *Proc. EUROCRYPT 2003*, pages 614–629. Springer-Verlag, 2003. LNCS 2656.
5. M. Bellare and C. Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In *Proc. ASIACRYPT 2000*, pages 531–545. Springer-Verlag, 2000. LNCS 1976.
6. M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *First ACM Conference on Computer and Communications Security*, pages 62–73, Fairfax, 1993. ACM.
7. M. Bellare and P. Rogaway. The exact security of digital signatures - how to sign with RSA and Rabin. In *Advances in Cryptology - Eurocrypt'96*, pages 399–416. Springer-Verlag, 1996. LNCS 1070.
8. C. Boyd and D. Park. Public key protocols for wireless communications. *The 1st International Conference on Information Secuirty and Cryptology (ICISC'98)*, pages 47–57, 1998.

9. X. Boyen. Multipurpose identity-based signcryption: A swiss army knife for identity-based cryptography. In *Proc. CRYPTO 2003*, pages 383–399. Springer-Verlag, 2003. LNCS 2729.
10. R. Canetti and H. Krawczyk. Analysis of key-exchange protocols and their use for building secure channels. In *Proc. EUROCRYPT 2001*, pages 453–474. Springer-Verlag, 2001. LNCS 2045. `http://eprint.iacr.org/2001/040/`.
11. D. Chaum and E. Van Heyst. Group signatures. In *Proc. EUROCRYPT 91*, pages 257–265. Springer-Verlag, 1991. LNCS 547.
12. R. Cramer, I. Damgård, and B. Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In *Proc. CRYPTO 95*, pages 174–187. Springer-Verlag, 1994. LNCS 839.
13. T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Trans. Inform. Theory*, 31:469–472, 1985.
14. O. Goldreich. *Foundations of Cryptography: Basic Tools*. Cambridge University Press, 2001.
15. S. Goldwasser, S. Micali, and R. Rivest. A digital signature scheme secure against adaptive chosen-message attack. *SIAM J. Computing*, 17(2):281–308, April 1988.
16. D. Harkins, C. Kaufman, and R. Perlman. The internet key exchange (IKE) protocol <draft-ietf-ipsec-ikev2-00.txt>. INTERNET-DRAFT, November 2001.
17. E. Van Herreweghen. Secure anonymous signature-based transactions. In *ESORICS '00: Proc. of the 6th European Symposium on Research in Computer Security*, pages 55–71. Springer-Verlag, 2000. LNCS 1895.
18. M. Jakobsson, K. Sako, and R. Impagliazzo. Designated verifier proofs and their applications. In *Proc. EUROCRYPT 96*, pages 143–154, 1996. LNCS 1070.
19. F. Laguillaumie and D. Vergnaud. Designated verifier signatures: Anonymity and efficient construction from any bilinear map. In *Proc. of the 4th Intl. Conference on Security in Communication Networks (SCN 2004)*, pages 105–119, 2004. LNCS 3352.
20. B. Lee, K. Kim, and J. Ma. Efficient public auction with one-time registration and public verifiability. In *Progress in Cryptology - INDOCRYPT 2001*, pages 162–174. Springer-Verlag, 2001. LNCS 2247.
21. Mastercard and Visa. *SET Secure Electronic Transactions Protocol Version 1.0*, May 1997. Available at http://www.setco.org/download.html.
22. NIST FIPS PUB 197. *Announcing the ADVANCED ENCRYPTION STANDARD (AES)*, November 2001.
23. R. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, February 1978.
24. R. Rivest, A. Shamir, and Y. Tauman. How to leak a secret. In *Proc. ASIACRYPT 2001*, pages 552–565. Springer-Verlag, 2001. LNCS 2248.
25. C. Schnorr. Efficient identification and signatures for smart cards. In *Proc. CRYPTO 89*, pages 239–252. Springer, 1990. LNCS 435.
26. G. Yang, D. Wong, and X. Deng. Analysis and improvement of a signcryption scheme with key privacy. In *Proc. of the 8th Information Security Conference (ISC '05)*, pages 218–232. Springer-Verlag, 2005. LNCS 3650.
27. G. Yang, D. Wong, and X. Deng. Efficient anonymous roaming and its security analysis. In *Proc. of the 3rd International Conference on Applied Cryptography and Network Security (ACNS 2005)*, pages 334–349. Springer-Verlag, 2005. LNCS 3531.