A Directed Signature Scheme Based on RSA Assumption

Rongxing Lu and Zhenfu Cao
(Corresponding author: Rongxing Lu)

Department of Computer Science and Engineering, Shanghai Jiao Tong University
No. 1954, Huashan Road, Shanghai, P. R. of China, 200030 (Email: {rxlu, cao-zf}@cs.sjtu.edu.cn)

(Received Aug. 13, 2005; revised and accepted Sep. 20 & Oct. 1, 2005)

Abstract

A directed signature scheme allows a designated verifier to directly verify a signature issued to him, and a third party to check the signature validity with the help of the signer or the designated verifier as well. Due to its merits, the directed signature scheme can be applied on some personally or commercially sensitive occasions. Up to now, there are several directed signature schemes having been proposed. However, to our best knowledge, none of them has provided the provable security proof. Therefore, in this paper, we would like to formally define the directed signature, and present a new directed signature scheme based on RSA assumption, then use the techniques from provable security to analyze the security of our proposed scheme.

Keywords: Digital signature, directed signature, provable security, RSA assumption

1 Introduction

Digital signature is one of the most important techniques in modern information security system for its functionality of providing data integrity and authentication. A normal digital signature [4, 8, 9] has the property that *anyone* having a copy of the signature can check its validity using the corresponding public information. This "self-authentication" property is necessarily required for some applications of digital signature such as official documents issued by some authorities. However, it is not suitable for some other applications, where a signed message is personally or commercially sensitive to the signature receiver, for example as in a bill of tax, a bill of health, etc. Therefore, to prevent potential misuse of signatures, it is preferable to place some restrictions on this property.

To achieve this purpose, Lim and Lee [5] first proposed the concept of directed signature at Auscrypto' 92. In a directed signature scheme, when a signer sends a signed message m to a designated verifier (receiver), then only the designated verifier can directly verify the signature on message m while the others know nothing on the origin and validity of the message m without the help of the signer or the designated verifier. On the other hand, if necessary, both the signer and the designated verifier can prove to any third party that the signature is a valid signature on the message m issued by the signer to the designated verifier. This property enables a dispute resolution in case that the signer tries to deny her signature or the designated verifier tries to deny the directedness of the signature.

In [5], Lim and Lee presented such a directed signature scheme based on GQ scheme [4]. Recently, other directed signature schemes based on Schnorr signature [9] also have been proposed [6, 7]. However, as we know, none of them has provided the provable security proof. On the other hand, there still does exist a directed signature scheme based on RSA assumption presently, though the ordinary RSA digital signature is very popular. Therefore, in this paper, we would like to formally define the directed signature scheme, and present a new directed signature scheme based on RSA assumption, then use the techniques from provable security to analyze its security. Our proposed directed signature scheme is converted by an ordinary RSA signature scheme, but it is fit for signing personally or commercially sensitive message and may be useful to the Internet community and Web-based systems community.

The rest of the paper is organized as follows. Section 2 contains some preliminaries about RSA problem, some notations and the formal definitions of directed signature scheme. Section 3 presents our proposed scheme, followed by the security analysis in Section 4. Finally, Section 5 concludes our paper.

2 Preliminaries

2.1 RSA Problem

Definition 1 (RSA Problem) *Let $n = p \cdot q$ be the product of two large primes of similar size and e, d be two integers such that $e \cdot d \equiv 1 \bmod \varphi(n)$, where $\varphi(n) = (p-1) \cdot (q-1)$. Given $n, e, y \in \mathbb{Z}_n^*$, compute the modular e-th root x of y such that $x^e = y \bmod n$. We define by $\mathsf{Succ}^{\mathsf{RSA}}_{\mathbb{Z}_n^*}(\mathcal{A})$ the success probability of an algorithm $\mathcal{A}$ in solving the RSA Problem as*

$$\mathsf{Succ}^{\mathsf{RSA}}_{\mathbb{Z}_n^*}(\mathcal{A}) = \Pr[\mathcal{A}(n, e, y = x^e \bmod n) = x \in \mathbb{Z}_n^*]$$

we say that the RSA assumption holds if $\mathsf{Succ}^{\mathsf{RSA}}_{\mathbb{Z}_n^*}(\mathcal{A})$ is negligible for any probabilistic polynomial time adversary $\mathcal{A}$.

2.2 Notations

We let $\mathbb{N} = \{1, 2, 3, \ldots\}$ be the set of positive integers. If x is a string, then $|x|$ denotes its length, while if $\mathbb{S}$ is a set then $|\mathbb{S}|$ denotes its size. If $k \in \mathbb{N}$ then 1^k denotes the string k ones. If $\mathbb{S}$ is a set then $s \xleftarrow{R} \mathbb{S}$ denotes the operation of picking a random element s of $\mathbb{S}$ uniformly.

We indicate that *Alice* is a signer, *Bob* is the designated signature verifier and *Carol* is any third party in the following scheme.

2.3 Framework of Directed Signature

A directed signature (DS) scheme consists of four algorithms: `Key Generation`, `Signature Generation`, `Signature Directed Verification`, and `Signature Public Verification`.

- `Key Generation (KG)`: On input of an unary string 1^k where k is a security parameter, it outputs a personal public and private key pair (pk, sk).
- `Signature Generation (SG)`: On input of a message m, a signer *Alice*'s public and private key pair (pk_A, sk_A), and a designated verifier *Bob*'s public key pk_B, it outputs a signature σ.
- `Signature Directed Verification (DV)`: On input of a purported signature σ, a message m, a signer *Alice*'s public key pk_A, and a designated verifier *Bob*'s public and private key pair (pk_B, sk_B), it outputs "accept" if (m, σ) is valid with respect to pk_A, and "reject" otherwise.
- `Signature Public Verification (PV)`: On input of a purported signature σ, a message m, a signer *Alice*'s public key pk_A, a designated verifier *Bob*'s public key pk_B, and a verifiable ***Aid*** provided by the signer *Alice* or the designated verifier *Bob*, it outputs "accept" if (m, σ) is valid with respect to pk_A, and "reject" otherwise.

These algorithms must satisfy the standard consistency constraint of the directed signature, *i.e.* if $\sigma = \mathtt{SG}(pk_A, sk_A, pk_B, m)$ and ***Aid*** is provided by *Alice* or *Bob*, $\mathtt{DV}(pk_A, pk_B, sk_B, m, \sigma) =$ accept, and $\mathtt{PV}(pk_A, pk_B, \boldsymbol{Aid}, m, \sigma) =$ accept.

The security of a directed signature scheme consists of two requirements: the *unforgeability* property and the *verifiable directedness* property. We say a directed signature scheme is secure if it satisfies two requirements.

Definition 2 (Unforgeability) *Let $\mathcal{A}$ be an adversary and Alice be a signer that involved in the following game.*

1) *$(pk_A, sk_A) \leftarrow \mathtt{KG}(1^k)$; where (pk_A, sk_A) is the public and private key pair of Alice.*
2) *$\mathcal{A}$ is given the public key pk_A of Alice, a designated verifier Bob's public and private key pair $(pk_B, sk_B) \leftarrow \mathtt{KG}(1^k)$, and allowed to make signing oracle query to Alice adaptively.*
3) *Finally, $\mathcal{A}$ outputs a signature $\sigma(m)$. $\mathcal{A}$ wins the game if $\sigma(m)$ is accepted. We define the success probability of $\mathcal{A}$ as*

$$\mathsf{Succ}^{\mathsf{EUF}}_{\mathsf{DS}}(\mathcal{A}) = \Pr\left[\begin{array}{l}(pk_A, sk_A) \leftarrow \mathtt{KG}(1^k);\\ \sigma(m) \leftarrow \mathcal{A}^{\mathcal{S}}(pk_A, pk_B, sk_B)\\ : \mathtt{DV}(pk_A, pk_B, sk_B, \sigma(m)) = accept.\end{array}\right].$$

We say a DS signature scheme is unforgeable if the probability $\mathsf{Succ}^{\mathsf{EUF}}_{\mathsf{DS}}(\mathcal{A})$ is negligible in the game.

Definition 3 (Verifiable Directedness) *We say a digital signature scheme is a DS scheme, if (i) only the designated verifier Bob can verify the authenticity of a purport signature issued to him; (ii) a third party Carol is able to verify a signature only with the help of the signer Alice or the designated verifier Bob.*

3 Our Proposed Scheme

In this section, based on the framework defined in Section 2.3, we will introduce our DS scheme based on RSA assumption.

Key Generation: For a given security parameter k, *Alice* first chooses two large prime p_a and q_a, where $|p_a| = |q_a| = k$, and computes $n_a = p_a \cdot q_a$ and $\varphi(n_a) = (p_a - 1) \cdot (q_a - 1)$. Then, she chooses a random odd number $e_a \in [1, \ldots, \varphi(n_a)]$ and computes d_a such that $e_a \cdot d_a \equiv 1 \pmod{\varphi(n_a)}$. At last, she keeps $sk_A = d_a$ as her private key and publishes the corresponding public key $pk_A = (n_a, e_a)$. Besides, *Alice* also publishes a secure one-way hash function H, where $H : \{0,1\}^* \rightarrow \mathbb{Z}_{n_a}^*$.

Similarly, *Bob* also first chooses two large prime p_b and q_b such that $|p_b| = |q_b| = k$, and computes $n_b = p_b \cdot q_b$ and $\phi(n_b) = (p_b - 1) \cdot (q_b - 1)$. Then, he chooses two parameters $e_b, d_b \in [1, \ldots, \varphi(n_b)]$ such

that $e_b \cdot d_b \equiv 1 \pmod{\varphi(n_b)}$. At last, he keeps d_b as his private key sk_B and publishes the corresponding public key $pk_B = (n_b, e_b)$.

Note that both $pk_A = (n_a, e_a)$ and $pk_B = (n_b, e_b)$ should be certified by a trusted authority.

Signature Generation Suppose that the signer *Alice* wants to send a signature of message m to a designated verifier *Bob* so that only *Bob* can directly verify it. *Alice* first chooses a random number $r \xleftarrow{R} \mathbb{Z}^*_{n_b}$, and computes R_1, R_2, where

$$R_1 = (r + m)^{e_b} \bmod n_b \quad (1)$$

$$R_2 = H(m, r)^{d_a} \bmod n_a \quad (2)$$

Then, *Alice* sends the signature $\sigma = (R_1, R_2)$, together with the message m, to the designated verifier *Bob*.

Signature Designated Verification Upon receiving $\sigma = (R_1, R_2)$ and m, the designated verifier *Bob* first uses his private key d_b to compute R'_1, where

$$R'_1 = {R_1}^{d_b} - m = (r + m)^{e_b d_b} - m = r \bmod n_b \quad (3)$$

Then, *Bob* checks the following equality

$$H(m, R'_1) = {R_2}^{e_a} \bmod n_a \quad (4)$$

If it does hold, $\sigma = (R_1, R_2)$ will be accepted.

Signature Public Verification In time of trouble or if necessary, either the signer *Alice* or the designated verifier *Bob* provides an ***Aid*** $R'_1 = r \in \mathbb{Z}^*_{n_b}$ to a third party *Carol*, *Carol* is then able to verify the signature $\sigma = (R_1, R_2)$ as follows,

First, *Carol* checks the validity of R'_1 by the following equality

$$R_1 = (R'_1 + m)^{e_b} \bmod n_b \quad (5)$$

If it holds, $R'_1 = r \in \mathbb{Z}^*_{n_b}$ will be accepted, otherwise rejected. Then, with valid R'_1, *Carol* can verify the signature $\sigma = (R_1, R_2)$ by Equation (4).

Clearly, if *Alice*, *Bob* and *Carol* all follow the issuing protocol, then from Equations (1)-(5), *Bob* and *Carol* always can verify the authenticity of a signature. Hence, the correctness follows.

4 Security

In this section, we will show that our proposed DS scheme satisfies the requirements stated in Section 2.3.

Theorem 1 *Let $\mathcal{A}$ be an adversary which can produce, with success probability ϵ, an existential forgery under chosen-message attacks [3] within a time τ, after q_h and q_s queries to the hash function H and the signing oracle respectively. Then the RSA problem can be resolved with another probability ϵ' within time τ', where*

$$\epsilon' \geq \frac{1}{(q_s + 1)\exp(1)} \cdot \epsilon$$
$$\tau' \leq \tau + (q_h + 2 \cdot q_s + 2) \cdot T_{\exp}$$

with $\exp(1)$ *the Napierian logarithm base and* $T_{\exp}$ *the time for an exponentiation evaluation.*

Proof: We now provide the proof of this theorem by the formalism introduced by Shoup [10, 11]. We define a sequence of games $\mathbf{G}_1$, $\mathbf{G}_2$, $\cdots$, of modified attack games starting from the actual game $\mathbf{G}_0$. Then, with these incremental games, we reduce a RSA problem instance (*i.e.*, given $n_a, e_a, y = x^{e_a} \bmod n_a$, compute such an $x \in \mathbb{Z}^*_{n_a}$) to an attack against the directed signature. We show that the adversary $\mathcal{A}$ can help us to resolve the RSA problem. GAME $\mathbf{G}_0$: This is an actual game, in the random oracle model [1]. The adversary $\mathcal{A}$ is allowed to access a random oracle $\mathcal{H}$ and a signing oracle $\mathcal{S}$. Moreover, the private-public key pair $(sk_B = d_b, pk_B = (n_b, e_b))$ of the designated verifier *Bob* is also available to $\mathcal{A}$.

To break the directed signature, the adversary $\mathcal{A}$ outputs its forgery, one then checks whether it is a valid signature or not. Note that the adversary $\mathcal{A}$ asks q_s queries to the signing oracle $\mathcal{S}$ and q_h queries to the random oracle $\mathcal{H}$, at most $q_s + q_h + 1$ queries are asked to the random oracle during this game, since each signing query may make such a new query, and the last verification step does too. We denote by Forge_0 the event that the forged signature is valid and use the same notation Forge_n in any game $\mathbf{G}_n$. By definition,

$$\epsilon = \mathsf{Succ}^{\mathsf{EUF}}_{\mathsf{DS}}(\mathcal{A}) = \Pr[\mathsf{Forge}_0]. \quad (6)$$

GAME $\mathbf{G}_1$: In this game, we will simulate the hash oracle $\mathcal{H}$ as usual by maintaining a hash list $\Lambda_{\mathcal{H}}$, the signing oracle $\mathcal{S}$ and the last verification step.

- For a hash query $\mathcal{H}(m, r \in \mathbb{Z}^*_{n_b})$, such that a record $(m, r, \perp, \perp, h)$ appears in $\Lambda_{\mathcal{H}}$, h will be responded to $\mathcal{A}$. Otherwise, a random number $h \xleftarrow{R} \mathbb{Z}^*_{n_a}$ is chosen, and a record $(m, r, \perp, \perp, h)$ will be added to $\Lambda_{\mathcal{H}}$. Note that the third and fourth components of the records of $\Lambda_{\mathcal{H}}$ will be explained in GAME $\mathbf{G}_2$.
- For a signing query $\mathcal{S}(m)$, one first chooses a random number $r \xleftarrow{R} \mathbb{Z}^*_{n_b}$, then asks for $h = \mathcal{H}(m, r)$ to the $\mathcal{H}$-oracle. The signature σ is then defined as $\sigma = \mathtt{SG}(pk_A, sk_A, pk_B, m)$.
- The game ends with the verification of the output $(m, \sigma = (R_1, R_2))$ for the adversary $\mathcal{A}$. One first uses $sk_B = d_b$ to compute $r = {R_1}^{d_b} - m \bmod n_b$ and asks for $h = \mathcal{H}(m, r)$, then checks whether $\mathtt{DV}(pk_A, pk_B, sk_B, m, \sigma) = \text{accept}$.

From the simulation above, the game is perfectly indistinguishable from the actual attack. Therefore,

$$\Pr[\mathsf{Forge}_1] = \Pr[\mathsf{Forge}_0]. \quad (7)$$

GAME $\mathbf{G}_2$: To implant the challenge $y = x^{e_a} \bmod n_a$ into hash answer, a real value α between 0 and 1 is introduced, which will be made precise later [2]. In this game, we modify the $\mathcal{H}$ oracle query, and leave other oracles unchanged.

- For a hash query $\mathcal{H}(m, r \in \mathbb{Z}^*_{n_b})$, such that a record (m, r, u, t, h) appears in $\Lambda_{\mathcal{H}}$, h will be responded to $\mathcal{A}$. Otherwise, one chooses a random number $u \xleftarrow{R} \mathbb{Z}^*_{n_a}$, and
  - with probability α, sets $h = u^{e_a} \bmod n_a$ and $t = 1$;
  - with probability $1 - \alpha$, sets $h = y \cdot u^{e_a} \bmod n_a$ and $t = 0$.

  Finally, a record (m, r, u, t, h) will be added to $\Lambda_{\mathcal{H}}$.

Because u is randomly chosen from $\mathbb{Z}^*_{n_a}$, then h is uniformly distributed in $\mathbb{Z}^*_{n_a}$, and this game is therefore perfectly indistinguishable from the previous one. Hence,

$$\Pr[\mathsf{Forge}_2] = \Pr[\mathsf{Forge}_1]. \tag{8}$$

GAME $\mathbf{G}_3$: In this game, we modify the signing oracle $\mathcal{S}$ query.

- For a signing query $\mathcal{S}(m)$, one first looks up (m, r, u, t, h) in $\Lambda_{\mathcal{H}}$.
  - If $t = 1$, one sets $\sigma = (R_1 = (r + m)^{e_b} \bmod n_b, R_2 = u \in \mathbb{Z}^*_{n_a})$ and returns it to $\mathcal{A}$. Obviously, $\sigma = (R_1, R_2)$ satisfies the verification Equations (3)-(4).
  - If $t = 0$, one terminates the game and reports failure.

Unless one signing query fails ($t = 0$) with probability $1 - \alpha$, this game is indistinguishable from the previous one. Therefore,

$$\Pr[\mathsf{Forge}_3] = \alpha^{q_s} \times \Pr[\mathsf{Forge}_2]. \tag{9}$$

By now, we have completed the simulation of hash oracle $\mathcal{H}$ as well as the signing oracle $\mathcal{S}$. Then, we try to use $\mathcal{A}$'s forgery $(m, \sigma = (R_1, R_2))$ to resolve the RSA problem. One looks up (m, r, u, t, h) in $\Lambda_{\mathcal{H}}$. In the found record,

- If $t = 1$, with probability α, one stops the game and reports failure.
- If $t = 0$, with probability $1 - \alpha$, one derives the challenge x by computing $\dfrac{R_2}{u} \bmod n_a$, since

$$\begin{aligned} & {R_2}^{e_a} = h = y \cdot u^{e_a} = (xu)^{e_a} \bmod n_a \\ \Rightarrow \quad & R_2 = xu \bmod n_a \Rightarrow x = \frac{R_2}{u} \bmod n_a \end{aligned}$$

As mentioned in GAME $\mathbf{G}_0$, there are at most $q_s + q_h + 1$ hash oracle queries, and each query requires one exponentiation evaluation, thus it costs $(q_s + q_h + 1)T_{\exp}$. In the signing query, each query also requires one exponentiation evaluation to compute $R_1 = (r + m)^{e_b} \bmod n_b$. Then, the time cost here is $q_s T_{\exp}$. Finally, in the verification step, another exponentiation evaluation is required for computing $R_2^{e_a} \bmod n_a$. Plus the time τ of running the adversary $\mathcal{A}$, the time τ' of one to resolve RSA problem is bounded by $\tau + (q_h + 2 \cdot q_s + 2) \cdot T_{\exp}$ in the end.

Therefore, based upon the analysis above, within the time $\tau + (q_h + 2 \cdot q_s + 2) \cdot T_{\exp}$, the success probability of one to resolve the RSA problem is

$$\mathsf{Succ}^{\mathsf{RSA}}_{\mathbb{Z}^*_n}(\tau + (q_h + 2 \cdot q_s + 2) \cdot T_{\exp}) = (1 - \alpha) \times \Pr[\mathsf{Forge}_3]. \tag{10}$$

From Equations (6), (7), (8), (9) and (10), we have

$$\begin{aligned} & \mathsf{Succ}^{\mathsf{RSA}}_{\mathbb{Z}^*_n}(\tau + (q_h + 2 \cdot q_s + 2) \cdot T_{\exp}) \\ = \ & (1 - \alpha) \times \Pr[\mathsf{Forge}_3] \\ = \ & (1 - \alpha) \times \alpha^{q_s} \times \Pr[\mathsf{Forge}_2] \\ = \ & (1 - \alpha) \times \alpha^{q_s} \times \Pr[\mathsf{Forge}_1] \\ = \ & (1 - \alpha) \times \alpha^{q_s} \times \Pr[\mathsf{Forge}_0] \end{aligned}$$

And then,

$$\begin{aligned} \mathsf{Succ}^{\mathsf{EUF}}_{\mathsf{DS}}(\mathcal{A}) \le & \frac{1}{(1 - \alpha) \times \alpha^{q_s}} \times \\ & \mathsf{Succ}^{\mathsf{RSA}}_{\mathbb{Z}^*_n}(\tau + (q_h + 2 \cdot q_s + 2) \cdot T_{\exp}) \end{aligned}$$

Since the maximum value of $(1 - \alpha) \times \alpha^{q_s}$ is $\dfrac{1}{q_s + 1} \cdot \left(\frac{1}{1 + \frac{1}{q_s}}\right)^{q_s}$, when $\alpha = \dfrac{q_s}{q_s + 1}$, we will have

$$\begin{aligned} \mathsf{Succ}^{\mathsf{EUF}}_{\mathsf{DS}}(\mathcal{A}) \le & (q_s + 1)\left(1 + \frac{1}{q_s}\right)^{q_s} \times \\ & \mathsf{Succ}^{\mathsf{RSA}}_{\mathbb{Z}^*_n}(\tau + (q_h + 2 \cdot q_s + 2) \cdot T_{\exp}) \end{aligned}$$

And for enough large q_s, $\left(1 + \frac{1}{q_s}\right)^{q_s} \approx \exp(1)$, the Napierian logarithm base, so we have

$$\begin{aligned} \mathsf{Succ}^{\mathsf{EUF}}_{\mathsf{DS}}(\mathcal{A}) \le & (q_s + 1)\exp(1) \times \\ & \mathsf{Succ}^{\mathsf{RSA}}_{\mathbb{Z}^*_n}(\tau + (q_h + 2 \cdot q_s + 2) \cdot T_{\exp}) \end{aligned}$$

This completes the proof. □

Theorem 2 *Our proposed scheme is really a directed signature scheme.*

Proof: To verify a signature $(\sigma = (R_1, R_2), m)$, $R_1' = {R_1}^{d_b} - m = r \bmod n_b$ must be available. Therefore, only the designated verifier *Bob* can verify its authenticity due to his private key d_b. As far as a third party *Carol* is concerned, to compute r from R_1 is equivalent to solve the RSA problem. However, when *Carol* holds r with the help of *Alice* or *Bob*, he can easily verify the signature. Hence, our proposed scheme is actually a directed signature scheme. □

5 Conclusions

The directed signature, due to its unforgeability and verifiable directedness properties, is very useful in some practical applications, where a signed message is personally or commercially sensitive. In this paper, we have formally defined the directed signature, then proposed a new directed signature scheme based on the RSA assumption and used the techniques from provable security to analyze the security of our proposed scheme.

Acknowledgment

The authors would like to thank anonymous reviewers for their valuable comments and suggestions that improve the presentation of this paper. This work is supported in part by the National Natural Science Foundation of China for Distinguished Young Scholars under Grant No. 60225007, the National Research Fund for the Doctoral Program of Higher Education of China under Grant No. 20020248024, and the Science and Technology Research Project of Shanghai under Grant Nos. 04JC14055 and 04DZ07067.

References

[1] M. Bellare and P. Rogaway, "Random oracles are practical: a paradigm for designing efficient protocols", in *Proc. of the 1st CCS*, ACM Press, New York, pp. 62–73, 1993.

[2] J. Coron, "On the exact security of full domain hash", in *Crypto'00*, LNCS 1880, pp. 229–235, Springer-Verlag, 2000.

[3] S. Goldwasser, S. Micali, and R. Rivest, "A digital signature scheme secure against adaptively chosen message attacks", *SIAM Journal on Computing*, vol. 17, no. 2, pp. 281–308, 1988.

[4] L. S. Guillou and J. J. Quisquater, "A practical zero-knowledge protocol fitted to security microprocessor minimizing both transmission and memeory", in *Eurocrypt' 88*, LNCS 403, pp. 216–231, Springer-Verlag, 1989.

[5] C. H. Lim and P. J. Lee, "Modified Maurer-Yacobi's scheme and its applictions", in *Auscrypt'92*, LNCS 2248, pp. 308–323, Springer-Verlag, 1992.

[6] S. Lan and M. Kumar, "A directed signature scheme and its applications", available at: http://arxiv.org/abs/cs/0409035.

[7] S. Lan and M. Kumar, "Some applications of directed signature scheme", available at: http://arxiv.org/abs/cs/0409050.

[8] R. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public key cryptosystems", *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[9] C. P. Schnorr, "Efficient signature generation for smart cards", *Journal of Cryptology*, vol. 4, no. 3, pp. 161–174, 1991.

[10] V. Shoup, "OAEP reconsidered", *Journal of Cryptology*, vol. 15, no. 4, pp. 223–249, Sep. 2002.

[11] V. Shoup, "Sequences of games: a tool for taming complexity in security proofs", available at: http://eprint.iacr.org/2004/332.

Rongxing Lu received his B.S. and M.S. degrees in computer science from Tongji University in 2000 and 2003 respectively. Currently, he is a doctoral candidate in the Department of Computer and Engineering, Shanghai Jiao Tong University. His research interests lie in cryptography and network security.

Zhenfu Cao is the professor and the doctoral supervisor of Computer Software and Theory at Department of Computer Science of Shanghai Jiao Tong University. His main research areas are number theory and modern cryptography, theory and technology of information security etc. He is the gainer of Ying-Tung Fok Young Teacher Award (1989), the First Ten Outstanding Youth in Harbin (1996), Best Ph.D thesis award in Harbin Institute of Technology (2001) and the National Outstanding Youth Fund in 2002.