A Concrete Certificateless Signature Scheme without Pairings

Aijun Ge, Shaozhen Chen
Department of Applied Mathematics
Zhengzhou Information Science and Technology Institute
Zhengzhou, Henan 450002, China
Email: geaijun@163.com, chenshaozhen@vip.sina.com

Xinyi Huang
School of Computer Science and Software Engineering
University of Wollongong
Wollongong, NSW 2500, Australia
Email: xyhuang81@gmail.com

***Abstract*—Certificateless public key cryptography was introduced to avoid the inherent key escrow problem in identity-based cryptography, and eliminate the use of certificates in traditional PKI. Most cryptographic schemes in certificateless cryptography are built from bilinear mappings on elliptic curves which need costly operations. Despite the investigation of certificateless public key encryption without pairings, certificateless signature without pairings received much less attention than what it deserves. In this paper, we present a concrete pairing-free certificateless signature scheme for the first time. Our scheme is more computationally efficient than others built from pairings. The new scheme is provably secure in the random oracle model assuming the hardness of discrete logarithm problem.**

***Keywords*-certificateless signature; bilinear pairing; discrete logarithm problem; random oracle model; strongly secure**

I. Introduction

In Asiacrypt 2003, Al-Riyami and Paterson proposed the notion of "Certificateless Public Key Cryptography" [1], whose original motivation is to find a public key system that does not use certificates, and at the same time does not have the key escrow problem. In certificateless cryptography, each user has two secrets, namely a secret value and a partial private key. The former is generated by the entity itself, and the latter is produced by a third party called as the Key Generating Center (KGC), who holds a master key. Decrypting or signing requires both secrets. As KGC does not know the secret value generated by the user, the key escrow problem is eliminated. The corresponding public key is generated by using the secret value and (optional) the partial private key. In certificateless cryptography, one's public key could be made available to other users by transmitting it along with messages (for example, in a signing application) or by placing it in a public directory (this would be more appropriate for an encryption setting).

Since the introduction of certificateless cryptography, a lot of schemes have been proposed so far(e.g., [4], [5], [6], [8], [9], [11]). However, as pointed out by Baek *et al.* [2], certificateless encryption schemes (and certificateless signature schemes) have been constructed within the framework of ID-based encryption proposed by Boneh and Franklin [3]. As a result, most certificateless cryptography schemes are based on bilinear mappings on elliptic curves, which are used to construct identity-based encryption and require heavy computational cost. Being aware of this, Baek *et al.* [2] proposed the first certificateless encryption scheme which does not depend on bilinear mappings. Sun *et al.* [12] improved their scheme and proposed a strongly secure certificateless encryption without pairings. Both schemes [2], [12] are more computationally efficient than others from bilinear mappings.

To the best of our knowledge, all concrete constructions of certificateless signatures in the literature are built from bilinear mappings. In this paper, we present the first concrete efficient certificateless signature scheme without pairings, and prove its security in the random oracle model. The security of our scheme can be reduced to discrete logarithm problem in finite fields. Our work is motivated by certificateless encryption schemes proposed in [2], [12]. Namely, by incorporating Schnorr signature [10] nontrivially, we obtain a certificateless signature scheme without pairings.

II. Preliminaries

This section reviews definitions of certificateless signatures and complexity assumptions associated with our scheme.

A. Syntax of Certificateless Signature Scheme

Definition 1: A certificateless signature scheme is made up of seven algorithms: **Setup**, **Partial-Key-Extract**, **Set-Secret-Value**, **Set-Public-Key**, **Set-Private-Key**, **Sign**, **Verify**. For a fixed security parameter k, these algorithms work as follows:

- **Setup**(k).
  This algorithm takes a security parameter k as input and returns the master secret key msk, the master public key mpk and a list of public system parameters $params$.
- **Partial-Key-Extract**$(params, ID, msk)$.
  This algorithm takes system parameters $params$, the master secret key msk and a user's identity ID as inputs, and returns a partial private key D_{ID} and (optional) a partial public key P_{ID} corresponding to the user with the identity ID.
- **Set-Secret-Value**$(params, mpk)$.
  This algorithm takes system parameters $params$ and the master public key mpk as inputs, and returns a secret value s_{ID}.
- **Set-Public-Key**$(params, mpk, ID, P_{ID}, s_{ID})$.
  This algorithm takes system parameters $params$, the

978-0-7695-3843-3/09 $26.00 © 2009 IEEE
DOI 10.1109/MINES.2009.100

master public key mpk, the user's identity ID, ID's partial public key P_{ID} and secret value s_{ID} as inputs, and returns the public key PK_{ID}.

- **Set-Private-Key**$(params, D_{ID}, s_{ID})$.
  This algorithm takes system parameters $params$, a user's partial private key D_{ID} and his/her secret value s_{ID} as inputs, and returns the private key SK_{ID}.
- **Sign**$(params, mpk, ID, SK_{ID}, m)$.
  This algorithm takes system parameters $params$, the master public key mpk, the user's identity ID, his/her private key SK_{ID} and the message m to be signed as inputs, and returns a certificateless signature σ.
- **Verify**$(params, mpk, ID, PK_{ID}, m, \sigma)$.
  This algorithm takes system parameters $params$, the master public key mpk, the user's identity ID, a public key PK_{ID}, and a message/signature pair (m, σ) as inputs, and returns "*valid*" or "*invalid*".

Completeness. For any correctly generated key pair (SK_{ID}, PK_{ID}),

$$\textbf{Verify}(params, mpk, ID, PK_{ID}, m, \textbf{Sign}(params, mpk,$$

$$ID, SK_{ID}, m)) = valid.$$

B. Security Model of Certificateless Signatures

As there is no certificate to authenticate a user's public key, it is reasonable to assume that an adversary can replace the user's public key with any value of its choice. Thus, two types of adversaries have been defined in certificateless cryptography [1]. A Type I adversary can replace any user's public key but does not have the partial private key of the target user, while a Type II adversary simulates a dishonest KGC who has the knowledge of the master secret key (and thus the partial private keys of all users), but is not allowed to replace the target user's public key.

For the security model of our certificateless signature scheme, we consider the strongest Type I adversaries defined in [5]: Super Type I adversary $\mathcal{A}_I$, which is given as much power as possible. $\mathcal{A}_I$ can obtain some message/signature pairs which are valid under the public key chosen by itself without providing the corresponding secret value.

For Type II adversary, we also consider the strongest adversary model "Super Type II adversary" defined in [5], which is given as much power as possible. $\mathcal{A}_{II}$ is allowed to obtain some message/signature pairs which are valid under the public key chosen by itself without providing the corresponding secret value. Note that $\mathcal{A}_{II}$ is not allowed to replace the target user's public key.

Definition 2: Let $Succ^{cma,cida}_{\mathcal{A}_I,super}$ be the success probability of a Super Type I adaptively chosen message and chosen identity adversary $\mathcal{A}_I$, a certificateless signature scheme is secure against Super Type I adversary $\mathcal{A}_I$ if $Succ^{cma,cida}_{A_1,super}$ is negligible for any polynomially bounded $\mathcal{A}_I$.

Definition 3: Let $Succ^{cma,cida}_{\mathcal{A}_{II},super}$ be the success probability of a Super Type II adaptively chosen message and chosen identity adversary $\mathcal{A}_{II}$, a certificateless signature scheme is secure against a Super Type II adversary $\mathcal{A}_{II}$ if $Succ^{cma,cida}_{\mathcal{A}_{II},super}$ is negligible for any polynomially bounded $\mathcal{A}_{II}$.

Remark. Due to page limitation, we refer interested readers to [5] for the formal game-based models of above definitions.

Definition 4: A certificateless signature scheme is existentially unforgeable against chosen message and chosen identity attacks if it satisfies Def. 2 and Def. 3.

C. Complexity Assumption

The security of our certificateless signature scheme can be reduced to the hardness of discrete logarithm problem. Let p, q be two primes and $q|(p-1)$. Let G be a subgroup of $\mathbb{Z}_p^*$ with prime order q and generator g. The discrete logarithm problem is defined as follows.

Definition 5: Given a random element $\beta \in G$, find $\alpha \in \mathbb{Z}_q$ such that $g^{\alpha} = \beta \pmod p$.

III. The Proposed Scheme

In this section, we describe our certificateless signature scheme without pairings. It consists of the following algorithms:

- **Setup:** This algorithm runs as follows:
  1) Given a security parameter $k \in \mathbb{N}$, this algorithm first chooses two primes p, q, where $p, q > 2^k$ and $q|(p-1)$. It then chooses an element $g \in \mathbb{Z}_p^*$ with order q. The subgroup generated by g is denoted as G;
  2) The master secret key x is randomly chosen from $\mathbb{Z}_q^*$, and the master public key is calculated as $y = g^x \pmod p$;
  3) Chooses three distinct cryptographic hash functions $H_1 : \{0,1\}^* \times \mathbb{Z}_p^* \to \mathbb{Z}_q^*$, $H_2 : \{0,1\}^* \times \mathbb{Z}_p^* \times \mathbb{Z}_p^* \to \mathbb{Z}_q^*$, $H_3 : \{0,1\}^* \times \{0,1\}^* \times (\mathbb{Z}_p^*)^4 \times \mathbb{Z}_q^* \times \mathbb{Z}_p^* \to \mathbb{Z}_q^*$.

  The system parameters *params*$=(p, q, g, G, y, H_1, H_2, H_3)$.
- **Partial-Key-Extract**: Given the user's identity $ID \in \{0,1\}^*$ as input, this algorithm works as follows:
  1) Picks $s_0, s_1 \in \mathbb{Z}_q^*$ at random, and calculates $p_0 = g^{s_0} \pmod p$ and $p_1 = g^{s_1} \pmod p$;
  2) Calculates $d_0 = s_0 + x \cdot H_1(ID, p_0) \pmod q$ and $d_1 = s_1 + x \cdot H_2(ID, p_0, p_1) \pmod q$;
  3) The partial private key $D_{ID} = d_0$, the partial public key $P_{ID} = (p_0, p_1, d_1)$.

  Remark: Algorithms **Setup** and **Partial-Key-Extract** are executed by KGC. Once the partial private and public keys are given to the user via secure channel, the user first checks if $g^{d_0} = p_0 \cdot y^{H_1(ID,p_0)} \pmod p$ and $g^{d_1} = p_1 \cdot y^{H_2(ID,p_0,p_1)} \pmod p$. If both equations hold, the user continues to run the following algorithms.
- **Set-Secret-Value**: This algorithm picks $z \in \mathbb{Z}_q^*$ at random, and sets $s_{ID} = z$ as the user's secret value.
- **Set-Private-Key**: Given the user's partial private key D_{ID} and the secret value s_{ID}, the full private key $SK_{ID} = (D_{ID}, s_{ID}) = (d_0, z)$.
- **Set-Public-Key**: Given the secret value s_{ID} and the partial public key $P_{ID} = (p_0, p_1, d_1)$, this algorithm

calculates $\mu = g^z \pmod p$. The user's public key $PK_{ID} = (P_{ID}, \mu) = (p_0, p_1, d_1, \mu)$.

- **Sign**: To sign a message $m \in \{0,1\}^*$, the signer
  1) Randomly selects $r, r' \in \mathbb{Z}_q^*$, and calculates $c = g^r \pmod p$ and $c' = g^{r'} \pmod p$);
  2) Sets $u = H_3(m, ID, c, c', PK_{ID})$;
  3) Calculates $v = r - uz \pmod q$ and $w = r' - ud_0 \pmod q$.

  The signature on the message m is $\sigma = (u, v, w)$.
- **Verify:** Given *params*, the signer's identity ID, $PK_{ID} = (p_0, p_1, d_1, \mu)$, a message m and the signature $\sigma = (u, v, w)$, the verifier checks if

$$g^{d_1} = p_1 y^{H_2(ID, p_0, p_1)} \pmod p$$

$$u = H_2(m, ID, g^v \mu^u, g^w (p_0 y^{H_1(ID, p_0)})^u, PK_{ID}).$$

  If both equations are correct, this algorithm outputs "*valid*". Otherwise, it outputs "*invalid*".

IV. Security Analysis

In this section, we will show that the proposed scheme is secure against Super Type I adversary and Super Type II adversary defined in Section II-B. This is ensured by the following two theorems.

Theorem 1: Our certificateless signature scheme is secure against Super Type I adversary in the random oracle model, assuming that the discrete logarithm problem is intractable on $\mathbb{Z}_p$.

Theorem 2: Our certificateless signature scheme is secure against Super Type II adversary in the random oracle model, assuming that the discrete logarithm problem is intractable on $\mathbb{Z}_p$.

The proofs of above theorems follow the same idea. We shall prove that if there is a Super Type I or Type II adaptively chosen message and chosen identity adversary which can break our certificateless signature scheme with non-negligible probability, then there exists another algorithm $\mathcal{B}$ which can solve the discrete logarithm problem with non-negligible success probability as well. Below we give the details of the proof of Theorem 1. Due to page limitation, we omit the proof of Theorem 2, which employs the similar technique to the proof of Theorem 1.

Proof of Theorem 1. Let $\mathcal{A}_I$ be a Super Type I adversary against our certificateless signature scheme. We want to build an algorithm $\mathcal{B}$ that uses $\mathcal{A}_I$ as a black-box to solve the discrete logarithm problem. At the beginning, $\mathcal{B}$ is given two primes p, q and a discrete logarithm problem instance $(g, \beta = g^\alpha)$. The goal of algorithm $\mathcal{B}$ is to find α such that $\beta = g^\alpha \pmod p$. We show that by acting as $\mathcal{A}_I$'s challenger, $\mathcal{B}$ can use $\mathcal{A}_I$ to find α.

$\mathcal{B}$ initializes $\mathcal{A}_I$ with the master public key $y = g^x$, where x is the master secret key and $\mathcal{B}$ keeps it secret. $\mathcal{B}$ then gives system parameters $params = (p, q, g, y, H_1, H_2, H_3)$ to A_I. Note that H_1 and H_2 are real hash functions, but H_3 is simulated by $\mathcal{B}$ as the random oracle.

At any time, $\mathcal{A}_I$ is allowed to access the following oracles in polynomial time. $\mathcal{B}$ responds to such queries as follows.

- **Create-User Request**: Suppose $\mathcal{A}_I$ makes at most q_{CU} queries to the **Create-User Request** oracle. At beginning, $\mathcal{B}$ chooses $t \in [1, q_{CU}]$ randomly. For the i^{th} **Create-User Request** query ID_i
  - If $i \neq t$, $\mathcal{B}$ picks $s_0, s_1, z_i \in \mathbb{Z}_q^*$ at random and computes $p_0 = g^{s_0} \pmod p), p_1 = g^{s_1} \pmod p$, $(d_0)_{ID_i} = s_0 + xe_{0i}$ and $(d_1)_{ID_i} = s_1 + xe_{1i}$, where $e_{0i} = H_1(ID_i, (p_0)_{ID_i})$, $e_{1i} = H_2(ID_i, (p_0)_{ID_i}, (p_1)_{ID_i})$. In this case, the corresponding partial public key of ID_i is P_{ID_i}= $((p_0)_{ID_i}, (p_1)_{ID_i}, (d_1)_{ID_i})$, $\mu_{ID_i} = g^{z_i} \pmod p$, the partial private key of ID_i is D_{ID_i} = $(d_0)_{ID_i}$, and the secret value s_{ID_i} = z_i.
  - Otherwise, $i = t$ and let $ID_i = ID^*$. $\mathcal{B}$ sets $p_0 = \beta = g^\alpha$ and $d_0 = \perp$, which means $\mathcal{B}$ cannot compute the partial private key of ID^*. $\mathcal{B}$ then picks $s_1, z_t \in \mathbb{Z}_q^*$ at random and calculates $(p_1)_{ID^*} = g^{s_1} \pmod p$, $(d_1)_{ID^*} = s_1 + xe_{1i}$, where $e_{1i} = H_2(ID^*, (p_0)_{ID^*}, (p_1)_{ID^*})$. In this case, the corresponding partial public key of ID^* is P_{ID^*} = $((p_0)_{ID^*}, (p_1)_{ID^*}, (d_1)_{ID^*})$, $\mu_{ID^*} = g^{z_t} \pmod p$, and the secret value $s_{ID^*} = z_t$.

  In either cases, $\mathcal{B}$ adds

$$(ID_i, D_{ID_i} = (d_0)_{ID_i}, s_{ID_i} = z_i,$$

$$PK_{ID_i} = ((p_0)_{ID_i}, (p_1)_{ID_i}, (d_1)_{ID_i}, (\mu)_{ID_i}))$$

  on list L and returns PK_{ID_i} to A_I
- **Partial-Private-Key Extraction**: When $\mathcal{B}$ receives a partial private key query for a created user ID_i,
  - If $ID_i = ID^*$, $\mathcal{B}$ returns "*failure*" and aborts the simulation.
  - Otherwise, $ID_i \neq ID^*$. $\mathcal{B}$ finds $(d_0)_{ID_i}$ on L and returns $(d_0)_{ID_i}$ as the answer.
- **Secret-Value-Extraction**: For the secret value extraction query on a created user ID_i, $\mathcal{B}$ finds s_{ID_i} on list L and returns it to $\mathcal{A}_I$ as the answer.
- **Public-Key-Replacement Request**: When A_{I} makes a public key replacement query on $\{ID_i, PK'_{ID_i} = ((p_0'), (p_1'), (d_1'), \mu')_{ID_i}\}$, $\mathcal{B}$ first checks whether

$$g^{(d_1')_{ID_i}} = (p'{}_1)_{ID_i} \cdot y^{H_2(ID, p_0', p_1')}.$$

  If the above equals, $\mathcal{B}$ returns "*failure*" and aborts the simulation. Otherwise, $\mathcal{B}$ replaces the original public key PK_{ID_i} with $(PK')_{ID_i}$. Then $\mathcal{B}$ will update the list L and rewrites the corresponding information as $(ID_i, D_{ID_i}, s_{ID_i}, (PK')_{ID_i})$. Note that the secret value and the partial private key corresponding to the new public key are not required.
- H_3 **Oracle Queries**: When $\mathcal{A}_I$ makes a query to oracle H_3, $\mathcal{B}$ first checks the list L_{H_3} to see if there is an entry for the same query. If $(m, ID, c, c', PK_{ID}, u)$ appears on L_{H_3}, then the same answer u will be given to $\mathcal{A}_I$.

Otherwise, a new random value u from $\mathbb{Z}_q^*$ will be given as the answer to $\mathcal{A}_I$. $\mathcal{B}$ then adds it to the list L_{H_3}.

- **Super-Sign Queries**: Suppose that $\mathcal{A}_I$ makes a signing query on (ID_i, m)
  - If $ID_i \neq ID^*$ and the public key of ID_i has not been replaced, $\mathcal{B}$ first finds the corresponding private key SK_{ID_i} = $((d_0)_{ID_i}, s_{ID_i} = z)$ on L. Then $\mathcal{B}$ uses the private key SK_{ID_i} to sign the message $\mathcal{B}$.
  - Otherwise, though $\mathcal{B}$ does not know the private key SK_{ID_i}, $\mathcal{B}$ chooses $(u, v, w) \in (\mathbb{Z}_q^*)^3$ at random and sets $u = H_3(m, ID, g^v(\mu)^u, g^w(p_0 y^{H_1(ID,p_0)})^u, PK_{ID})$. If the collision happens, $\mathcal{B}$ will rechoose (u, v, w) until there is no collision happens on L_{H_3}.

  In either case, $\mathcal{B}$ outputs $\sigma = (u, v, w)$ as ID_i's signature on m. As the random oracle H_3 is controlled by $\mathcal{B}$, each signature $\sigma = (u, v, w)$ will pass the verification. The above simulated signature is identically distributed as the one in the real attack. By doing this, $\mathcal{B}$ performs a perfect simulation.

Eventually, $\mathcal{A}_I$ outputs a valid signature $(ID, m, \sigma = (u, v, w))$.

- If $ID \neq ID^*$ or σ is not a valid signature, $\mathcal{B}$ returns "*failure*".
- Otherwise, $\mathcal{B}$ can solve the discrete logarithm problem by applying the forking technique.

According to the forking lemma [7], if $\mathcal{A}_I$ is a sufficient efficient forger in the above interactions, $\mathcal{B}$ can obtain two valid signatures $\sigma = (u, v, w)$ and $\sigma' = (u', v', w')$ $(u \neq u')$ that satisfies

$$g^w(p_0 y^{H_1(ID,p_0)})^u = g^{w'}(p_0 y^{H_1(ID,p_0)})^{u'}$$

$\mathcal{B}$ can calculate $\alpha = \log_g \beta = \log_g(p_0) = (w' - w)/(u - u') - xH_1(ID, \beta)$.

Probability of Success: It remains to compute the probability that $\mathcal{B}$ solves the given instance of the discrete logarithm problem. $\mathcal{B}$ succeeds if:

1) Λ_1: $\mathcal{B}$ does not abort during the simulation;
2) Λ_2: $(ID, m, \sigma = (u, v, w))$ can pass the verification under the current public key PK_{ID};
3) Λ_3: In the forgery $(ID, m, \sigma = (u, v, w))$, $ID = ID^*$. This happens with probability $1/q_{CU}$.

$\mathcal{B}$ does not abort during the simulation if and only if the following events happen:

1) Λ_{11}: $\mathcal{A}_I$ does not make **Partial-Private-Key Extraction** request of ID^*. Suppose that $\mathcal{A}_I$ makes at most q_{PPK} queries to the oracle **Partial-Private-Key Extraction**, this happens with probability $(1 - \frac{1}{q_{CU}})^{q_{PPK}}$.
2) Λ_{12}: A_I does not make a **Public-Key-Replacement** that satisfies

$$g^{(d_1')_{ID_i}} = (p'_1)_{ID_i} \cdot y^{H_2(ID,p_0',p_1')}$$

According to forking lemma in [7], if A_I finds another (p_0', p_0', d_1', u') that satisfies $g^{(d_1')_{ID_i}} = (p'_1)_{ID_i} \cdot y^{H_2(ID,p_0',p_1')}$, then the discrete logarithm problem can been solved with probability $\varepsilon \geq 7Q/q$, in polynomially bounded time, where Q is the number of queries that A_I can ask to the random oracle H_2. This will not happen as discrete logarithm is assumed to be hard on $\mathbb{Z}_p$.

Therefore, the probability that $\mathcal{B}$ can solve the discrete logarithm problem is

$$Adv_B^{DL} \geq \frac{1}{q_{CU}}(1 - \frac{1}{q_{CU}})^{q_{PPK}} Succ_{A_I,\sup er}^{cma,cida}$$

V. Conclusion

In this paper, we present the first concrete certificateless signature scheme without pairings. Our construction is motivated by certificateless encryption schemes without pairings proposed in [2], [12]. The new scheme is provably secure (in the random oracle model) against Super Type I and Super Type II adversaries defined in [5], assuming that the discrete logarithm problem is intractable. The proposed scheme is more computationally efficient than other certificateless signature schemes from bilinear mappings.

Acknowledgment

The authors would like to express their gratitude to Qiong Huang, Yinxia Sun and the anonymous reviewers for their valuable suggestions and comments.

This paper is supported by the National Natural Science Foundation of China (NO. 60673081).

References

[1] Al-Riyami, S.S., Paterson, K.G.: Certificateless Public Key Cryptography. In: Laih, C.S. (ed.) ASIACRYPT 2003. LNCS, vol. 2894, pp. 452-473. Springer, Heidelberg (2003)

[2] Baek, J., Safavi-Naini, R., Susilo, W.: Certificateless Public Key Encryption without Pairing. In: Zhou, J., Lopez, J., Deng, R.H., Bao, F.(eds.) ISC 2005. LNCS,vol. 3650, pp. 134-148. Springer, Heidelberg (2005)

[3] Boneh, D., Franklin, M.: Identity-based Encryption from the Weil Pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213-229. Springer, Heidelberg (2001)

[4] Barbosa, M., Farshim, P.: Certificateless signcryption. Proceedings of the 2008 ACM symposium on Information, computer and communications security,pp. 369-372. (2008)

[5] Huang, X., Mu, Y., Susilo, W., Wong, D.S., Wu, W.: Certificateless Signature Revisited. In: Pieprzyk, J., Gnodosi, H., Dawson, E.(eds.) ACISP 2007, LNCS, vol. 4586, pp. 308-322. Springer, Heidelberg (2007)

[6] Huang, Q., Wong, D.S.: Generic Certificateless Encryption in the Standard Model. In: Atsuko Miyaji, Hiroaki Kikuchi and Kai Rannenberg.(eds.) IWSEC 2007, LNCS, vol. 4752, pp. 278-291. Springer, Heidelberg (2007)

[7] Pointcheval, D., Stern, J.: Security Arguments for Digital Signature and Blind Signature. Journal of Cryptology 13(3), 361-396 (2000)

[8] Rafael, C., Ricardo, D.: Two Notes on the Security of Certificateless Signatures. In: W. Susilo, J.K. Liu, Y. Mu. (eds.) ProSec 2007, LNCS, vol. 4784, pp. 85-102. Springer, Heidelberg (2007)

[9] Raylin, Tso, Xun, Yi, Huang X.Y.: Efficient and Short Certificateless Signature. In: M. K., Franklin, L. C. K., Hui, Wong, D. S. (eds.) CANS 2008, LNCS, vol. 5339, pp. 64-79. Springer, Heidelberg (2008)

[10] Schnorr, C.P.: Efficient Signature generation for Smart Cards. Joural of Cryptology 4(3), 239-252 (1991)

[11] Sherman S.M.Chow, Colin Boyd, Juan, M.G.N.: Security Mediated Certificateless Signature. In: M. Yung et al.(eds.) PKC 2006, LNCS, vol. 3958, pp. 508-524. Springer, Heidelberg (2006)

[12] Sun, Y.X., Zhang, F.T., Baek, Joonsang.: Strongly Secure Certificateless Public Key Encryption Without Pairing. In: F. Bao et al. (eds.) CANS 2007, LNCS, vol. 4856, pp. 194-208. Springer, Heidelberg(2007)