A new proxy identity-based signcryption scheme for partial delegation of signing rights

Hassan Elkamchouchi and Yasmine Abouelseoud
Eelctrical Engg. Dept., Faculty of Eng., Alexandria University, Alexandria, Egypt
**email address: yasmine.abouelseoud@gmail.com*

In this paper, a new identity-based proxy signcryption scheme is presented. The proposed scheme allows partial delegation of signing rights by granting the proxy agent a private key that is extracted from that of the principal signer. Consequently, a signature created by the proxy signer is distinguishable from that created by the principal signer. This level of security is a common requirement in many applications to prevent malicious proxy agents from impersonating the principal signer. In the identity-based setting, the public key of a user is derived from his identity, thus simplifying certificates management process compared to traditional public key cryptosystems. Moreover, the scheme is based on bilinear pairings over elliptic curves and thus smaller key sizes are required compared to schemes not utilizing elliptic curves. A revocation protocol of dishonest agents is given together with a renewal procedure for the proxies of honest agents. Finally, an application scenario for the proposed scheme is presented.

في هذا البحث، تم اقتراح طريقة تشفير و إمضاء مدمج بالوكالة في ظل نظام تشفير يعتمد على الهوية. الطريقة المقترحة تسمح بالتوكيل الجزئي و ذلك عن طريق منح الوكيل مفتاح خاص مستنتج من المفتاح الخاص للعميل الأصلي. و بناء على ذلك، فإن توقيع العميل الأصلي يختلف عن توقيع الوكيل. هذا المستوى من الأمان مطلوب في كثير من التطبيقات لمنع الوكيل من انتحال شخصية العميل الأصلي. في نظم التشفير المعتمدة على الهوية، فإن المفتاح العام لأي مستخدم يكون مستنبط من هويته و بالتالي يسهل عملية إدارة الشهادات عنه في نظم تشفير المفتاح العام التقليدية. علاوة على ذلك، فإن الطريقة المقترحة تعتمد على المنحنيات الناقصية مما يسمح باستخدام مفاتيح أصغر. كما تم اقتراح طريقة لاستبعاد الوكلاء الغير أمناء مع تجديد المفاتيح الخاصة بالوكلاء الآخرين. في النهاية، تم عرض سيناريو لتطبيق الطريقة المقترحة.

Keywords: Proxy signcryption, Identity-based cryptography, Network security

1. Introduction

The large scale adoption of computing and network technologies for carrying out on-line transactions and message transmissions have been greatly supported by the research and advances in the area of cryptography and network security. The cryptographic primitives such as encryption and digital signatures are used to build protocols that provide specific security services such as transmission of a message over an insecure channel while protecting the integrity and confidentiality of the message contents. Signcryption [1] is a public key cryptographic primitive that combines the functionalities of a digital signature and encryption in a single logical step. Consequently, it achieves reductions in computations and communications overhead.

Proxy signcryption [2] has been introduced as a practical cryptographic solution to the problem of secure and authenticated message transmission by a networked computer with low computational capacity. Many widely used personal communications devices such as digital assistants, hand-held computers, pagers and mobile phones belong to this category. The low computational capability constraint models the lack of hardware features in these devices to efficiently carry out the heavy mathematical computations required by cryptographic primitives such as digital signatures. Therefore, proxy signature schemes [3-4] have emerged to allow off-loading of heavy computational work from a low power device to a more powerful server.

Identity-based cryptosystems, first introduced by Shamir [5], are becoming common those days. The basic idea is to get rid of public key certificates by allowing the

user's public key to be the binary sequence corresponding to an information identifying him in a non-ambiguous way (e-mail address, social security number,...). This kind of system allows to avoid trust problems encountered in certificate based Public Key Infrastructures (PKIs): there is no need to bind a public key to its owner's identity since those are one single thing. These systems involve trusted authorities called Private Key Generators (PKGs) whose task is to compute users' private keys from their identity information (users do not generate their key pairs themselves). Several identity-based signcryption schemes have been introduced in the literature. Few examples include the work in [6- 8].

In this paper, a new proxy identity-based signcryption scheme for partial delegations of signing rights is presented. Partial delegation schemes are of special interest due to the security they offer and the lower computational and communications costs associated with them compared to other types of delegations. The proposed scheme is validated and its performance related issues are addressed. A revocation protocol for dishonest proxies is also provided.

The rest of the paper is organized as follows. The next section offers an overview of proxy delegation schemes. Section 3 gives the general description of an identity-based signcryption scheme. In Section 4, the proposed proxy identity-based signcryption scheme is presented along with the necessary mathematical background. In Section 5, the consistency of the proposed scheme is validated and its performance is analyzed. Section 6 provides an application scenario for the proposed scheme. Finally, Section 7 concludes the paper.

2. Overview of proxy delegation schemes

A proxy signature [3] allows a designated person, called a proxy signer/agent, to sign on behalf of a principal signer. In other words, a proxy signature allows a user to delegate his signing rights to a designated signer. There are different types of delegations: full delegation, partial delegation and delegation by warrant.

Full Delegation: In full delegation schemes, a proxy signer is given the same secret SK that the principal signer has, so that both of them create the same signatures. Obviously, when the proxy signer deliberately signs a document unfavorable to the principal signer, this misbehavior is not detected because the signature created by the proxy signer is indistinguishable from the signatures created by the principal signer.

Partial Delegation: In partial delegation schemes, a new secret SK^* is created from SK, thus leading to the modification of the verification equation and SK^* is given securely to the proxy signer. The proxy signature is checked by the modified equation and not by the original equation. This implies that a signature created by the proxy signer is distinguishable from a signature created by the principal signer. In such delegation schemes, only the public key of the principal signer is required for the signature verification.

Delegation by Warrant: This sort of delegation is implemented by using a warrant, which certifies that the proxy is exactly the signer to be entrusted. Delegation by warrant is performed by the consecutive execution of the signing phase of the public key signature scheme in use. There are two approaches to signature schemes for this type. In the first approach, a warrant consists of a message part and a principal signer's signature for a public key of the designated proxy signer. Given the warrant, the proxy signer signs a document under his secret key by an ordinary signature scheme. In the second approach, a warrant consists of a message part and a principal signer's signature for a newly generated public key for the designated proxy signer. The secret key compatible with this generated public key is given to the proxy signer in a secure way.

Proxy signature schemes have been constructed for each of these types of delegations and the most adequate scheme should be selected depending on the user's need for security, message length and the signer's and/or verifier's computational capabilities. The partial delegation and delegation by warrant are more secure than the full delegation. This is in a sense that created proxy signatures are distinguishable from ordinary signatures. It is noteworthy that

partial delegation schemes are computationally more efficient than delegations by warrant, especially in the signature verification phase. Moreover, the message length is shorter for partial delegations than in case of using delegations by warrant. Consequently, this motivates further research in partial delegation schemes.

3. Identity-based signcryption

Since Zheng introduced signcryption as a cost effective public key primitive that achieves both confidentiality and authenticity, it has received significant attention from many researchers. A formal security model has been developed and other variants of the original scheme by Zheng have been proposed [9]. One interesting variant is the adaptation of the basic signcryption scheme to the identity-based setting [7-8]. Any identity-based signcryption scheme consists of the following four algorithms:

- *Setup:* Given a security parameter *k*, the Private Key Generator (PKG) generates the system public parameters *params*.
- *Key Generation:* Given an identity *ID*, the PKG computes the corresponding private key d_{ID} and transmits it to its owner in a secure way.
- *Signcrypt:* In order to send a message *m* to *B*, the sender *A* computes $Signcrypt(m, d_{ID_a}, ID_b)$ to obtain the corresponding ciphertext σ.
- *Unsigncrypt:* When the recipient *B* receives σ, he computes $UnSigncrypt(\sigma, d_{ID_b}, ID_a)$ and obtains the plaintext *m* or the symbol $\perp$ if σ was an invalid ciphertext.

The consistency condition must be satisfied, that is:

$$m = UnSigncryp\,t\ (Signcrypt(m, d_{ID_a}, ID_b), d_{ID_b}, ID_a)$$

4. The proposed proxy identity-based signcryption scheme

In this section, the proposed proxy identity-based signcryption scheme is presented. First, the necessary mathematical background is reviewed for completeness.

4.1. Elliptic curves

An elliptic curve E [10] over a finite field F_p is defined by the equation

$$y^2 = x^3 + ax^2 + bx + c$$

where $D = a^2b^2 - 4a^3c - 4b^3 + 18abc - 27c^2 \neq 0$ and $x \in F_p$. One of the popular elliptic curves used in cryptography is given by $y^2 = x^3 + 1$ over F_p, where p is a prime satisfying that $p \equiv 2 \mod 3$ and $p = lq - 1$, where q is also a prime. Let G_1 be a subgroup of points in $E(F_p)$ of order q. The popularity of the curve stems from the ease of embedding binary sequences onto points on that curve [11]. Let $y_0 \in F_p$ be the binary sequence to be embedded:

1. Compute $x_0 = (y_0{}^2 - 1)^{1/3} = (y_0{}^2 - 1)^{(2p-1)/3} \in F_p$
2. Let $Q = (x_0, y_0) \in E(F_p)$ and set $Q_{ID} = lQ \in G_1$
3. utput Q_{ID}.

4.2. Bilinear Pairings

Many efficient identity-based encryption and signature schemes in the literature are based on the use of bilinear pairings, which are briefly defined below [12].

Consider two groups G_1 (additive) and G_2 (multiplicative) of the same prime order q. A bilinear map $\hat{e} : G_1 \times G_1 \to G_2$ satisfying the following properties is needed.

- *Bilinearity:* $\forall\ P, Q \in G_1, \forall\ a, b \in F_q^*$, we have that $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$,
- $\hat{e}(P, Q + R) = \hat{e}(P, Q)\ \hat{e}(P, R)$.
- *Non-degeneracy:* For any point $P \in G_1$, we have $\hat{e}(P, Q) = 1$ for all $Q \in G_1$ iff $P = \mathrm{O}$
- *Computability:* There exists an efficient algorithm to compute $\hat{e}(P, Q), \forall\ P, Q \in G_1$.

The modified Weil pairing and the Tate pairing [11] are admissible instantiations. The modified Weil pairing works on the elliptic

curve defined by $y^2 = x^3 + 1$, considered above. In this case, G_1 is a cyclic subgroup of the additive group of points on that supersingular elliptic curve $E(F_p)$ over a finite field. G_2 is a cyclic subgroup of the multiplicative group associated to a finite field extension of F_p.

4.3. The proposed key extraction procedure for the proxy agent

Let the public key of the principal agent be the point $Q_{ID} \in G_1$ and the corresponding private key be the point $d_{ID} = sQ_{ID}$, where s is the master secret key of the PKG. Also, let $P_{pub} = sP$ where P is the generator of the group G_1. In order to generate the public/private key pair of the proxy agent the two parties follow the steps given below.

1. The principal agent chooses a random element $x \in F_q^*$ and computes $U = xP$ and publishes it.
2. The private key of the designated proxy agent is computed by the principal agent as $d_{proxy} = d_{ID} + xP_{pub}$.
3. The private key is then securely transmitted to the proxy agent.
4. The proxy agent validates his private key by checking that the following condition holds.

$$\hat{e}(P, d_{proxy}) = \hat{e}(P_{pub}, Q_{ID} + U)$$

If the private key is properly constructed, it must satisfy the above condition. To see this, consider the following argument.

$$\begin{aligned}\hat{e}(P, d_{proxy}) &= \hat{e}(P, d_{ID} + xP_{pub}) \\ &= \hat{e}(P, sQ_{ID} + xsP) \\ &= \hat{e}(sP, Q_{ID} + xP) \\ &= \hat{e}(P_{pub}, Q_{ID} + U)\end{aligned}$$

5. Finally, the proxy agent accepts the key pair $(Q_{ID_{pro}}, d_{proxy}) = (Q_{ID} + U, d_{proxy})$ as his public/ private key pair. It is noteworthy that $d_{proxy} = d_{ID} + xP_{pub} = sQ_{ID} + xsP = sQ_{ID} + sU$,th us $d_{proxy} = sQ_{ID_{pro}}$.

4.4. The Proposed proxy identity-based signcryption scheme for partial delegation of signing rights

Zheng [1] showed how to use a shortened version of the digital signature standard to build an efficient signcryption scheme. B. Libert and J. Quisquater [7] recently showed that Hess's ID-based signature [13] can also be used as a building block to obtain a provably secure identity-based signcryption scheme which relies on the hardness of the Decisional Bilinear Diffie-Hellman (DBDH) problem. The proxy version of the scheme in [7] is presented below. It consists of the following procedures.

Setup: Given security parameters *k* and *n*, the PKG chooses the system parameters that include two groups G_1 and G_2 of prime order q $(q > 2^k)$, a bilinear map $\hat{e} : G_1 \times G_1 \to G_2$, a generator P of G_1, a master secret $s \in_R F_q^*$ and a public key $P_{pub} = sP \in G_1$. PKG also chooses a secure symmetric cipher (E, D), and secure hash functions

$$H_1 : \{0,1\}^* \to G_1,\ H_2 : G_2 \to \{0,1\}^n$$

and

$$H_3 : \{0,1\}^* \times G_2 \to F_q.$$

The public parameters are

$$< G_1, G_2, \hat{e}, P, P_{pub}, H_1, H_2, H_3 >$$

Key Generation: Given an identity *ID*, the PKG computes $Q_{ID} = H_1(ID) \in G_1$ and the corresponding private key $d_{ID} = sQ_{ID} \in G_1$.

Proxy Key Generation: For a principal agent whose public/private key pair is (Q_{ID}, d_{ID}), a random element $x \in F_q^*$ is chosen. He publishes $U = xP$ and the proxy public/private key pair $(Q_{ID} + U, d_{ID} + xP_{pub})$ is transferred to the proxy agent, who validates it as illustrated above.

Signcrypt by the Proxy Agent: To send a message m to B on behalf of A, the proxy agent follows the steps below

1. Compute $Q_{ID_B} = H_1(ID_B) \in G_1$
2. Choose $\alpha \leftarrow_R F_q^*$ and compute $k_1 = \hat{e}(P, P_{pub})^{\alpha}$ and $k_2 = H_2(\hat{e}(P_{pub}, Q_{IDB})^{\alpha})$.
3. Compute $c = E_{k_2}(m)$, $r = H_3(c, k_1)$ and $S = \alpha P_{pub} - rd_{proxy} \in G_1$. The ciphertext is $\sigma = (c, r, S, U)$. The only difference in this proxy version of the scheme in [7] is that the proxy agent signs on behalf of the principal agent A using the proxy private key given to him by A.

Unsigncrypt a ciphertext produced by proxy agent: Upon receiving $\sigma = (c, r, S, U)$, B performs the following tasks

1. Compute $Q_{ID_A} = H_1(ID_A) \in G_1$
2. Compute $Q_{ID_{pro}} = Q_{ID_A} + U$
3. Compute $k_1 = \hat{e}(P, S)\, \hat{e}(P_{pub}, Q_{ID_{pro}})^r$
4. Compute $\tau = \hat{e}(S, Q_{ID_B})\ \hat{e}(Q_{ID_{pro}}, d_{ID_B})^r$ and $k_2 = H_2(\tau)$.
5. Recover $m = D_{k_2}(c)$ and accept σ iff $r = H_3(c, k_1)$

The intended recipient may be some proxy agent for B instead of B itself. The only difference in this case is that the public/private keys of the recipient are replaced by those of the proxy agent.

5. Consistency and performance assessment of the proposed scheme

In this section, the consistency of the proposed scheme is validated and performance related issues are addressed. Moreover, a proxy revocation protocol is presented.

5.1. Consistency validation

The consistency is easy to verify by the bilinearity of the map.

$$\begin{aligned}\hat{e}(P,S)\hat{e}(P_{pub}, Q_{ID_{pro}})^r &= \hat{e}(P, \alpha P_{pub} - rd_{proxy})\ \hat{e}(sP, rQ_{ID_{pro}}) \\ &= \hat{e}(sP, \alpha P - rQ_{ID_{pro}})\ \hat{e}(sP, rQ_{ID_{pro}}) \\ &= \hat{e}(P_{pub}, \alpha P) = \hat{e}(P_{pub}, P)^{\alpha} = k_1\end{aligned}$$

and

$$\begin{aligned}\hat{e}(S, Q_{ID_B})\hat{e}(Q_{ID_{pro}}, d_{ID_B})^r &= \hat{e}(\alpha P_{pub} - rd_{proxy}, Q_{ID_B}) \\ &\quad \times \hat{e}(rd_{proxy}, Q_{ID_B}) \\ &= \hat{e}(\alpha P_{pub}, Q_{ID_B}) \\ &= \hat{e}(P_{pub}, Q_{ID_B})^{\alpha} = \tau\end{aligned}$$

5.2. Performance evaluation

It is clear that the proxy key generation phase involves two scalar multiplications and one point addition in the group G_1 at the principal agent's side having low computational capabilities. Both operations are far less expensive than pairing evaluations involved in the signature generation phase. The validation process of the proxy key at the proxy agent's side involves two pairing evaluations and one point addition. However, the proxy agent is assumed to have much better computational power compared to the principal agent. Moreover, the proxy key generation/validation is carried out once and the results may be safely stored afterwards. Furthermore, it is noteworthy that the verification process only requires the knowledge of the public key of the principal agent.

Compared to the scheme in [7], the proposed proxy version involves an extra point addition operation and the ciphertext is expanded by appending U to it. Thus, the delegation of signing rights is achieved with low additional costs. Thus, the proposed scheme is more efficient than other schemes where the delegation is done by warrant [14-15].

The scheme provides the non-repudiation property. Any third party (like firewalls) can be convinced of the message origin by recovering k_1 just like in step 3 of unsigncryption given above and checking if the condition $r = H_3(c, k_1)$ holds. The knowledge of the plaintext m is not required for the public verification of the origin of a ciphertext.

The computational efficiency of the above scheme may be greatly enhanced if enough storage is available to the communicants. In this case, frequently communicating parties

may pre-compute the pairings $\hat{e}(P, P_{pub})$, $\hat{e}(P_{pub}, Q_{ID_{pro}})$ and $\hat{e}(Q_{ID_{pro}}, d_{ID_B})$ before hand.

5.3. Proxy revocation

A proxy agent might give his proxy to others or frequently signs malicious messages, the principal agent should revoke proxies of dishonest proxy agents in order to stop further abuses. There are two simple ways to revoke the proxy agent's signing capability [3].

- To make a revocation list publicly seen. When a problem occurs, the principal agent puts U or $Q_{ID_{pro}}$ in the list. Every verifier checks this list at first and if no corresponding entry is found in the list, the verification process starts.
- To change the public key of the principal agent and accordingly update all proxies of honest proxy agents. In the identity-based setting changing the public key is achieved by choosing a different identifier ID' for the principal agent, this in turn incurs the change of the public key derived as $Q_{ID} = H_1(ID')$.

In both revocation mechanisms the revocation information, updated revocation list or the new public keys, must be widely propagated to all potential verifiers of a proxy signature.

For the second type of revocation an honest proxy agent needs to have a new proxy after the renewal of the public key. As described above, a designated proxy agent has already been given a proxy of the old public key in a secure way. By the following protocol, an honest proxy agent can update his proxy through an insecure channel.

New Public Key Creation: A principal agent *A* selects a new identifier ID', may be by appending the current date to his identifier, and sends it to the PKG. The PKG then computes the corresponding private key $d_{ID'} = sH_1(ID')$ and transfers it to the principal agent in a secure way. *A* publishes the new public key $Q_{ID'} = H_1(ID')$ and keeps $d_{ID'}$ secret.

Identification: After this announcement, a proxy agent who wants to update his proxy asks the principal agent to send him a new proxy. To this end, the proxy agent proves his identity by some identification protocol [16].

New Proxy Creation: After the principal agent is convinced of the proxy agent's identity, the principal agent looks for the old secret proxy variable x in his secret proxy variable list. He calculates the old proxy $d_{proxy} = d_{ID} + xP_{pub}$ and the new proxy given by $d'_{proxy} = d_{ID'} + x'P_{pub}$, where x' is a randomly chosen element and $U' = x'P$. Finally, he computes $\bar{d}_{proxy} = d'_{proxy} - d_{proxy}$ and sends $(\bar{d}_{proxy}, U')$ to the proxy agent.

New Proxy Construction: Using the received information, the proxy agent calculates $d'_{proxy} = d_{proxy} + \bar{d}_{proxy}$ and checks its validity by $\hat{e}(P, d'_{proxy}) = \hat{e}(P_{pub}, Q_{ID'} + U')$.

Thus, by means of the above protocol, the renewal process may be carried out over an insecure channel.

6. A Practical application scenario

The proposed proxy signcryption scheme may be used to construct a proxy agent based communication protocol for secure message transmission. The protocol described below assumes that the principal and the proxy agent have already agreed on a session key k_s through some key agreement identity-based protocol [17]. Both the principal and the proxy agent perform the proxy key generation step to generate, transfer and verify the proxy secret. The following steps are in order.

1. The principal agent uses the shared session key and symmetric encryption to transmit the ciphertext $C = < E_{k_s}(m), hash(k_s, m) >$ to the proxy agent.

2. The proxy agent decrypts *C* to recover the message *m* and verifies its integrity using the keyed hash function. The proxy agent then uses the signcryption step to create the proxy-signcrypted ciphertext $C_{proxy}(m)$.

3. The cryptogram is transmitted to its destination over a public channel.

4. In the reverse process, the proxy-agent receives a proxy-signcrypted or signcrypted message addressed to its end-point principal.

5. Assuming a proxy-signcrypted message, the proxy-agent uses the unsigncryption step to recover and authenticate the message. If

the message recovery action is successful, then the proxy agent uses some session key k_s^* and symmetric key encryption to prepare the message for transmission onwards to the principal agent.

6. The principal agent receives the incoming message from the proxy agent and decrypts it to recover the plaintext message.

The above protocol relieves the principal agent from all pairing evaluations. However, if the message is intended for the principal agent-- and not his proxy agent-- two extra pairing evaluations need to be computed but still due to public verifiability of the scheme the proxy agent can authenticate the message for the principal agent.

7. Conclusions

In this paper, a new proxy identity-based signcryption scheme is presented. The scheme allows partial delegation of signing rights and thus offers a desirable level of security against malicious proxy agents. The use of proxy agents relieves principal agents, which are usually networked devices with low computational capacity, from the burden of mathematical computations required for message authentication.

Signcryption is a cryptographic primitive that achieves both message confidentiality and origin authentication in a single logical step and thus introduces reductions in computations. Consequently, the proposed proxy signcryption scheme may be used as a building block for an efficient and secure communications protocol involving devices with low computational capabilities. A revocation protocol for dishonest proxy agents is also provided along with a renewal procedure for proxies of honest ones.

References

[1] Y. Zheng, "Signcryption and Its Applications in Efficient Public Key Solutions", In Proceedings of ISW'97, pp. 291-312 (1998).

[2] C. Gamage, J. Lewiwo and Y. Zheng, "An Efficient Scheme for Secure Message Transmission using Proxy-Signcryption", In Proceedings of the 22nd Australian Computer Science Conference, Auckland, New Zealand (1999).

[3] M. Mambo, K. Usuda and E. Okamoto, "Proxy Signatures for Delegating Signing Operation", In Proceedings of the 3rd ACM Conference on Computer and Communications Security (CCS) (1996).

[4] Boldyreva, A. Palacio and B. Warinschi, "Secure Proxy Signature Scheme for Delegation of Signing Rights", In IACR Eprint Archive, Available at http://eprint.iacr.org/2003/096/.

[5] Shamir, "Identity Based Cryptosystems and Signature Schemes", In Advances in Cryptology- CRYPTO 1984, Springer-Verlag, LNCS 0196 (1984).

[6] X. Boyen, "Multi-Purpose Identity-Based Signcryption (A Swiss Army Knife for Identity-Based Cryptography)", In Proceedings of CRYPTO 2003, pp. 383-399 (2003).

[7] Libert and J. Quisquater, "New Identity Based Signcryption from Pairings", In Proceedings of IEEE Information Theory Workshop (2003).

[8] J. Malone-Lee, "Identity Based Signcryption", available at http://eprint.iacr.org/2002/072/.

[9] J. Baek, R. Steinfeld and Y. Zheng, "Formal Proofs for the Security of Signcryption", In Proceedings of PKC'02, 2002, Springer LNCS 2274.

[10] J.H. Silverman, "The Arithmetic of Elliptic Curves", GTM 106, Springer-Verlag (1986).

[11] Boneh and M. Franklin, "Identity Based Encryption from the Weil Pairing", In Advances in Cryptology- CRYPTO 2001, Springer, LNCS 2139 (2001).

[12] Joux, "A One-Round Protocol for Tripartite Diffie-Hellman Algorithm", Number Theory Symposium- ANTS-IV, Springer-Verlag, LNCS 1838, pp. 385-394 (2000).

[13] F. Hess, "Efficient Identity Based Signature Schemes Based on Pairings", In Proceedings of SAC, Springer, LNCS Series (2002).

[14] S. Duan, Z. Cao and Y. Zhou, "Secure Delegation by Warrant Identity-Based Proxy Signcryption Scheme", In Proceedings of Computational

Intelligence and Security Conference CIS 2005, LNAI 3802, Springer-Verlag (2005).

[15] Q. Wang and Z. Cao, "Efficient ID-Based Proxy Signature and Proxy Signcryption from Bilinear Pairings", In Proceedings of Computational Intelligence and Security Conference CIS 2005, LNAI 3802, Springer-Verlag (2005).

[16] K. Kurosawa and S.H. Heng, "From Digital Signature to ID-Based Identification/Signature", In Proceedings of Public Key Cryptography- PKC'04, LNCS 2947, Springer-Verlag (2004).

[17] N. McCullagh and P.S.L.M Barreto, "A New Two-Party Identity-Based Authenticated Key Agreement", In Proceedings of CT-RSA 2005, Also Available Online at http://eprint.iacr.org/2004/122.

Received February 18, 2008
Accepted October 30, 2008