CONSTRUCTING A PROXY SIGNATURE SCHEME BASED ON EXISTING SECURITY MECHANISMS

Wei-Bin LEE and Tzung-Her CHEN

Introduction

Due to the growth of the Internet, e-commerce is widespread and the security of Internet transactions is a matter that is becoming more and more important and challenging. Fortunately, the digital signature and the digital time stamp are well-defined tools used to address this challenge. Digital signature schemes are widely used in security mechanisms such as integrity, authentication and non-repudiation. They can be used to check the integrity of a message, authenticate the origin, and protect from dishonest repudiation. Digital time stamp schemes are used to ascertain when digital data were created or when data were signed.

However, a conventional digital signature is not suitable for some practical applications. For example, a team leader wants to take a trip to a tourist attraction where there is no computer network to use. Hence, during his vacation, he must delegate to a trusted member of his staff to perform his tasks including signing electronic documents. However, conventional digital signature schemes do not address the proxy function, and it is not reasonable to give the secret signing key to the proxy. To provide a solution, the proxy signature scheme was proposed in 1996.[1,2] The proxy signature allows a designated person, called a proxy signer, to sign a message on behalf of an original signer. Many proxy signature schemes have been proposed. Unfortunately, there are still permanent challenges, such as security and complexity, in the proposed schemes. Mambo, Usuda and Okamoto describe a situation,[3] where it was possible for the original signer to forge a proxy signature on behalf of the proxy signer, a situation called repudiation.[4] Sun and Hsieh argue that Mambo and coworkers' proxy signature scheme has a delegation transfer problem.[5] This means that the proxy signer can transfer the proxy without both the agreement and the consciousness of the original signer. Therefore, another party can generate a "valid" proxy signature on behalf of the original signer. Later, certain nonrepudiable

INFORMATION & SECURITY. An International Journal, Vol.12, No.2, 2003, 250-258.

proxy signature schemes [6,7,8] and threshold proxy signature schemes[9,10,11,12] were proposed. The reader may refer to a number of references for details.[13,14,15,16,17,18,19,20,21]

Actually, the strength of a cryptographic scheme cannot really be proved. When a new scheme is proposed, the authors always believe that their scheme is strong, secure, and unbreakable if one does not know the secret key. In fact, all that the authors can do is to demonstrate the scheme's power against some known attacks. However, we often find that there is always a new attack invented for a new scheme; hence, a newly proposed scheme almost always suffers from some inborn weakness, so we must always be careful when applying a new cryptographic scheme. To reduce this concern, a novel proxy signature scheme is proposed that does not invent a new mathematical model, but rather combines well-defined tools and existing mechanisms, such as the digital signature and the time stamp to satisfy the requirements of proxy signature. The scheme can be implemented by conventional digital signature schemes and public key infrastructures without significant modifications. Therefore, unknown security problems introduced by a new mathematical model can be minimized.

This paper is organized as follows. In the section that follows, the authors briefly introduce the Mambo-Usuda-Okamoto scheme and some other well-designed cryptographic tools and mechanisms. Then, a novel proxy signature scheme is proposed. Security analysis and discussions are given after that.

Preliminaries

Review of the Mambo-Usuda-Okamoto Scheme

To understand the concept of the proxy signature scheme, a brief review of the Mambo-Usuda-Okamoto scheme is necessary.[22]

Denote $s \in Z^*_{p-1}$ as a private key of an original signer and $v = g^s \bmod p$ as the corresponding public key, where p is a prime and g is a generator for Z^*_p.

Step 1. Proxy generation

An original signer generates a random number $k \in Z^*_{p-1}$ and computes $K = g^k \bmod p$. Furthermore, he determines $\sigma = s + kK \bmod p\text{-}1$.

Step 2. Proxy delivery

The original signer delivers the proxy (σ, K) to a proxy signer over a secure channel.

Step 3. Proxy verification

The proxy signer checks for congruence as to whether or not $g^{\sigma} = vK^{K} \bmod p$. If the equation holds, the proxy signer accepts it as a valid proxy.

Step 4. Signing by the proxy signer

When the proxy signer signs a message *m* on behalf of the original signer, he uses the σ as an alternative to *s*, and executes the ordinary signing operation. Thus, (*m*, (Signature of the original scheme), *K*) serves as a created proxy signature.

Step 5. Verification of the proxy signature

The verification of the proxy signature is the same as in the ordinary signature scheme except for the extra computation $v\,K^{K} \bmod p$, which is dealt with as a new public value.

There are six main security properties to be satisfied by a proxy signature scheme: unforgeability, secret-key's dependence, verifiability, distinguishability, identifiability and undeniability.[23] These properties are discussed in detail below.

Roles of Certification Authority and Time Stamping Authority

In general, the digital signature operation signs a message using a private key. Subsequently, anyone can verify it using the corresponding public key. However, the challenge of how to ascertain who really owns the public key has arisen. To ascertain the genuine public key, the accepted solution is to make a trusted party, called a Certification Authority (CA), digitally sign data structures. This is known as certification – mapping between public key and identity information. If someone knows CA's public key, he can ascertain that the public key belongs to a particular person.

On the other hand, digital time stamp schemes are used to ascertain when a particular event took place, for example, when digital data were created, a digital message was sent or received, a digital signature was generated or a signature key was revoked/overdue.[24] In order to associate a message with a particular time, a Time Stamping Authority (TS) has been standardized by IETF. Furthermore, it is well known that time stamping plays an important role in digital signature schemes. According to Zhou and Lam, "A typical approach to secure digital signatures as non-repudiation evidence relies on the existence of an on-line trusted time-stamping authority (TS). Each newly generated digital signature has to be time-stamped by a TS so that the trusted time of signature generation can be identified."[25]

The Proposed Proxy Signature Scheme

The following notations are used to represent message and protocols in this paper:

ID_U: identity information of party U.

S_U and V_U: the private key and the corresponding public key of party U.

$sS_A(m)$: digital signature of message m with the private key S_A.

$A \rightarrow B$: X: party A delivers message X to party B.

There are several participants involved in this scenario, including an original signer (for example, a manager), a proxy signer (for example, a secretary), CA, and TS. Each party has a regular key pair, certificated by CA, including TS's (S_{TS}, V_{TS}), the original signer's (S_o, V_o) and the proxy signer's (S_p, V_p). For example, the manager goes on vacation for one week. He creates a temporary proxy-signature key pair (s_p, v_p) based on the same cryptographic assumption. Subsequently, the delegation information, including a proxy-signature key, is delivered to TS for time stamping. After receiving the time-stamped delegation information, the signing and verifying operations of a proxy signature are the same as in existing ordinary digital signature schemes. The detailed steps are given as follows:

Step 1. Proxy generation

The original signer designates a proxy signer and generates a short-term key pair (s_p, v_p) for the proxy signer. The expiry date Td of the delegation should also be defined. Furthermore, the delegation message is determined by creating the signature $D = sS_o(ID_o, ID_p, v_p, Td)$,
Original signer → TS: ID_o, ID_p, v_p, Td, D
TS: verifying the validity of D with V_o
TS → Original signer: Tt, $sS_{TS}(D, Tt)$, where Tt denotes the timestamp.
The original signer verifies the validity of $sS_{TS}(D, Tt)$.

Step 2. Proxy delivery

The original signer sends $(ID_o, ID_p, (s_p, v_p), Td, D, Tt, sS_{TS}(D, Tt))$ to the proxy signer over a secure channel.

Step 3. Proxy verification

The proxy signer authenticates the proxy signature key s_p with the public key v_p, then checks the validity of D and $sS_{TS}(D, Tt)$, if necessary. Thus the expiry date Td and the delegation relationship between the origin signer and the proxy signer are confirmed. It is worth emphasizing that s_p is a temporary and short-term key.

Step 4. Proxy signature generation

The proxy signer generates the proxy signature of a message *m* with the signature key s_p based on an ordinary digital signature scheme. Thus the signing operation generates (*m*, (Signature of the original scheme)). Finally, (*m*, (Signature of the original scheme), ID_o, ID_p, v_p, *Td*, *D*, *Tt*, $sS_{TS}(D, Tt)$) serves as generated proxy signature.

If necessary, the proxy signer could sign the signature again with his individual private key to prevent a malicious original signer from forging a proxy signature on behalf of the proxy signer.

Step 5. Verification of the proxy signature

The verification of the proxy signature is divided in two phases. The first phase checks whether or not the proxy signature is valid. This is the same as the procedure for the ordinary signature scheme. The second phase checks the validity of the expiry date *Td* and the proxy relationship between the original signer and the proxy signer. This is achieved by checking the validity of the signatures *D* and *sSTS(D, Tt).*

Security Analysis and Discussion

The proposed proxy signature scheme is straightforward and easy to implement based on the currently existing public-key infrastructure. Due to the fact that the security of the signature is inherent in the original scheme, the delegation process causes the major security concern. Therefore, it is worthwhile to further discuss the role that TS plays in the proposed scheme. It is known that the proxy signature schemes focus on the security issue of the temporary proxy-signature key pairs. In the proxy signature schemes, a proxy-signature key is a short-term key and it is only valid during a specified period. However, CA is responsible to issue and maintain the certification of the regular keys, i.e., the long-term keys, including their creation and revocation. Nevertheless, short-term keys demand minimal key management and protection. It is inappropriate and impractical for a CA to confirm these short-term keys. For the sake of reducing cost, TS, instead of CA, issues the certificate for the proxy key by time-stamping the delegation information and the expiry date. Appending a timestamp by TS is more economical than generating a regular certificate by CA. Discussions related to the security and the advantages of the proposed scheme are given in the following sub-sections.

Discussion of Essential Properties

The paper discusses the following properties that have to be satisfied by a proxy signature scheme:

1. Unforgeablity: It is impossible for anyone to create a valid proxy signature without knowing the private key s_p.
2. Secret-key's dependence: The original signer using his certificated private key signs a proxy signature key. It implies that the proxy signature key is computed from the secret key of the original signer.
3. Verifiability: Anyone can verify the validity of a proxy signature using the corresponding public key, verified by CA.
4. Distinguishability: Anyone can verify the proxy signature by the proxy signing key *vp* which is generated by the original signer with his individual private key V_o. That is to say that a verifier can distinguish a proxy signature from the regular signature signed by the original signer.
5. Identifiability: The verifier can determine the relationship of delegation between an original signer and a proxy signer by verifying the delegation message *D*. Hence, the verifier can determine the corresponding proxy signer from a proxy signature.
6. Undeniability: Due to the fact that the delegation information is signed by the original signer and timestamped by TS, a proxy signer can not deny his behavior.

Other Properties

The proposed scheme is based on the security of existing cryptographic tools and commercial products. Therefore, any attack to forge a valid proxy signature will fail unless an adversary can defeat sophisticated security mechanisms. There are still some properties that cause concern.

1. In the proxy generation phase, the original signer signs the delegation information as the proxy certificate, $sS_o(ID_o, ID_p, v_p, Td)$, which is subsequently appended to the proxy signature. Therefore, the delegation relationship between the original signer and the proxy signer is addressed and proved. Hence, it is impossible for the signer to transfer the proxy without the agreement of the original signer. This property can avoid the delegation transfer problem. Furthermore, the alternative of additionally signing the message with his individual private key can further overcome this problem, unless the proxy signer releases his private key. Meanwhile, the original signer cannot forge a valid proxy signature. That is, the proxy signer cannot claim that the proxy signature in dispute is illegally signed by the original signer, i.e., non-repudiation.
2. Based on well-defined commercial TS and CA mechanisms,[26,27,28] the proposed scheme naturally has fewer security considerations.

3. All necessary mechanisms have already been implemented in the real world, so the proposed scheme can be easily implemented without any problems.
4. The proxy signature key automatically expires when the expiry date arrives. There is no extra cost to maintain or revoke the proxy signature keys. Furthermore, because the proxy signature key is a temporary and short-term key, there are fewer security problems than with regular keys.

Furthermore, in order to prevent a malicious proxy signer from signing even if the expiry date arrives, the proxy signature must be time-stamped by TS. This is a general secure digital signature problem, and therefore is omitted here.

Conclusions

The authors apply the currently existing CA and TS mechanisms in a straightforward way to construct a solution to the problem of the security challenges of newly proposed proxy signature-related schemes. The proposed proxy signature scheme not only satisfies the essential properties mentioned in the Mambo-Usuda-Okamoto's proxy signature scheme but also has the following additional advantages: it provides non-repudiation and prevents delegation transfer. It is obvious that the scheme does not affect the current security infrastructure and, thus, is more practical than the previously proposed schemes.

Notes:

[1] Masahiro Mambo, Keisuke Usuda, and Eiji Okamoto, "Proxy Signatures: Delegation of the Power to Sign Messages," *IEICE Transactions on Fundamentals* E79-A, 9 (1996): 1338-1354.

[2] Masahiro Mambo, Keisuke Usuda, and Eiji Okamoto, "Proxy Signatures for Delegating Signing Operation," in *Proceeding of the 3rd ACM Conference on Computer and*

Communications Security (New Delhi, India, March 14-15, ACM Press New York, 1996), 48-57.

[3] Ibid.

[4] Kan Zhang, "Threshold Proxy Signature Schemes" (paper presented at the Information Security Workshop, Japan, September 1997), 191-197.

[5] Hung-Min Sun and B.-T. Hsieh, "Remark on Two Nonrepudiable Proxy Signature Schemes," in *Proceedings of the 9th National Conference on Information Security* (Taiwan, 1999), 241-246.

[6] Hung-Min Sun, "An Efficient Nonrepudiable Threshold Proxy Signature Scheme with Known Signers," *Computer Communications* 22, 8 (May 1999): 717-722.

[7] Sun and Hsieh, "Remark on Two Nonrepudiable Proxy Signature Schemes."

[8] Hung-Min Sun, Narn-Yih Lee, and Tzonelih Hwang, "Nonrepudiable Threshold Proxy Signatures," in *Proceedings of the 9th National Conference on Information Security* (Taiwan, 1999), 254-261.

[9] Sun, "An Efficient Nonrepudiable Threshold Proxy Signature Scheme with Known Signers."

[10] Sun, Lee, and Hwang, "Nonrepudiable Threshold Proxy Signatures."

[11] Hung-Min Sun, Narn-Yih Lee, and Tzonelih Hwang, "Threshold Proxy Signatures," *IEE Proceedings- Computers and Digital Techniques* 146, 5(1999), 259-263.

[12] Kan Zhang, "Threshold Proxy Signature Schemes."

[13] Shin-Jia Hwang and Chi-Hwai Shi, "Specifiable Proxy Signature Schemes," in *Proceedings of the National Computer Symposium*, (Taiwan, 1999), C-190-197; Shin-Jia Hwang and Chi-Hwai Shi, "A Simple Multi-Proxy Signature Scheme," in *Proceedings of the 10th National Conference on Information Security*, (Taiwan, 2000), 134-138.

[14] Seungjoo Kim, Sangjoon Park, and Dongho Won, "Proxy Signatures, Revisited," in *Information and Communications Security (ICICS'97)*, ed. Yongfei Han, Tatsuaki Okamoto, Sihan Qing, LNCS 1334, (Berlin: Springer-Verlag, 1997), 223-232.

[15] W.-B. Lee and C.-Y. Chang, "Efficient Proxy-Protected Proxy Signature Scheme Based on Discrete Logarithm," in *Proceedings of the 10th National Conference on Information Security*, (Hualien, Taiwan, 2000), 4-7.

[16] Mambo, Usuda, and Okamoto, "Proxy Signatures for Delegating Signing Operation."

[17] Sun, "An Efficient Nonrepudiable Threshold Proxy Signature Scheme with Known Signers;" Hung-Min Sun, "Convertible Proxy Signature Scheme," in *Proceedings of the National Computer Symposium* (1999), C-186-189; Sun and Hsieh, "Remark on Two Nonrepudiable Proxy Signature Schemes."

[18] Hung-Min Sun and Biing-Jang Chen, "Time-Stamp Proxy Signature with Traceable Receivers," in *Proceedings of the 9th National Conference on Information Security*, (Taiwan, 1999), 247-253.

[19] Sun, Lee, and Hwang, "Nonrepudiable Threshold Proxy Signatures."

[20] Sun, Lee, and Hwang, "Threshold Proxy Signatures."

[21] Zhang, "Threshold Proxy Signature Schemes."

[22] Mambo, Usuda, and Okamoto, "Proxy Signatures: Delegation of the Power to Sign Messages;" Mambo, Usuda, and Okamoto, "Proxy Signatures for Delegating Signing Operation."

[23] Mambo, Usuda, and Okamoto, "Proxy Signatures for Delegating Signing Operation."

[24] Jianying Zhou and Kwok-Yan Lam, "Securing Digital Signatures for Non-Repudiation," *Computer Communications* 22, 8 (May 1999): 710-716.

[25] Ibid.

[26] Digistamp, <http://www.digistamp.com> (14 January 2004).

[27] SSE, <http://www.trustedweb.com> (14 January 2004).

[28] Surety, <http://www.surety.com/> (14 January 2004).

WEI-BIN LEE received his B.S degree from the Department of Information and Computer Engineering, Chung-Yuan Christian University, Chungli, Taiwan, in 1991 and his M.S. degree in Computer Science and Information Engineering from the National Chung Cheng University, Chiayi, Taiwan, in 1993. He received his Ph.D. degree in 1997 from National Chung Cheng University. Since 1999, he has been with the Department of Information Engineering at the Feng Chia University, where he is currently an Associate Professor. His research interests currently include cryptography, information security, steganography and data security. He is an honorary member of the Phi Tau Phi Scholastic Honor Society. *Address for correspondence:* Department of Information Engineering, Feng Chia University, Taiwan.
E-mail: lwb@ iecs.fcu.edu.tw

TZUNG-HER CHEN received his B.S. degree from the Department of Information & Computer Education from the National Taiwan Normal University, Taiwan, Republic of China, in 1991, and a M.S. degree from the Department of Information Engineering, Feng-Chia University, in 2001. He is currently pursuing his Ph.D. degree in the Department of Computer Science, National Chung-Hsing University. His research interests include cryptography, information hiding and digital watermarking. *Address for correspondence:* Department of Computer Science, National Chung-Hsing University, 250 Kuo-Kuang Road, Taichung 40227, Taiwan R.O.C. *FAX:* 886-4-22853869 *E-mail:* phd9007@cs.nchu.edu.tw