Digital Multisignature Schemes for Authenticating Delegates in Mobile Code Systems

Shiuh-Pyng Shieh, *Senior Member, IEEE*, Chern-Tang Lin, Wei-Bon Yang, and Hung-Min Sun

***Abstract*—In this paper, we motivate the need for efficient multisignature schemes in delegated mobile services. With the schemes, delegates can be identified and delegated accesses can be controlled. First, we give a new digital signature scheme with message recovery. Based on the digital signature scheme, two digital multisignature schemes are proposed: the parallel multisignature scheme and the serial multisignature scheme. The parallel multisignature scheme allows each user to sign the same message separately and independently, and then combines all individual signatures into a multisignature. The serial multisignature scheme allows a group of users to sign the message serially, and does not need to predetermine the signing order. Both multisignature schemes can withstand the attacks that aim to forge the signatures or to get the private keys of the signers.**

***Index Terms*—Authentication, digital signature, management of mobile services, security.**

I. Introduction

With the growth of the Internet, many services are provided to help network users to access remote data and codes. The development of World Wide Web (WWW) combines many traditional services and allows users to navigate the entire Internet using a single Web browser [1]. The debut of WWW motivates the need of mobile codes and data. The mobile code can be transmitted across networks and executed on the other end, and therefore can help users acquire many services over the Internet in a more effective way. The future trend of WWW is to integrate the mobile stations with the Internet such that mobile users can still acquire the Internet services while traveling. With limited resources in hand, both mobile devices and clients are likely to rely heavily on remote servers through the use of mobile codes and data. In this scheme, a code or data object is transmitted over networks from a remote server to a mobile device only when it is needed. In this way, the memory requirement of mobile devices is greatly reduced. The Java(TM) language is a simple, object-oriented, portable, robust language that supports mobile codes [5]–[7], [28]. Java augments the present WWW capabilities by dynamically downloading the mobile code fragments, called applets, and running these code fragments locally. Since the mobile codes are transmitted across insecure networks from possibly untrusted sources and executed in the local browser, it raises serious security issues [4]. Therefore, in the design of mobile services on the Internet, the security problem is considered to be an important one. No one wants to bring across any piece of code if there is a possibility that executing the code could 1) damage any hardware, software, or information on the host machine; or 2) pass unauthorized information to anyone [29].

The paper intends to provide a possible solution for the security problems in WWW where the mobile codes and data of the current version still cannot be authenticated. As the latest report JDK1.1 from JavaSoft states [14], the new JavaSecurity API will support digital signatures to authenticate classes, images, sounds, and other types of data. A new javakey tool can generate digital signatures for Java files. With the digital signature scheme, its access control mechanism can be constructed. JDK 1.2 also aims to provide built-in primitives to support basic concepts in secure distributed computing, such as authentication and delegation.

With mobile codes and data, mobile service systems allow users to navigate the entire information space, where every piece of information is connected via links to related pieces of information [24]. A user who has access to an object can activate all links to other objects. Therefore, it is necessary to differentiate the access privileges for the requests from different access paths. In Mosaic 2.0 and NCSA httpd [16], it is possible to restrict access to the information contained in a directory to specific hosts or authenticated users [25], [27], and consequently the traversal will fail if the user does not have the authorization. This approach does not protect the relationships between objects. To cope with the authorization problem, Samarati *et al.* proposed a model [24] which takes into consideration the relationships among linked objects, and allows administrative privileges to be delegated. Consider the case that a service is partitioned among multiple servers, a client can issue a request package, which may contain mobile code and data fragments, to the delegates. Upon receiving the request package, the delegate may perform the service, and then forward the request package to the next delegate. The receiving delegate repeats the same process. In this sequence of operations, user's privilege to access an object through different delegates over different link paths is different, and the delegates that participate in the computation must be authenticated. That is, both the client issuing the request package and the delegates forwarding it must sign the request so that the requesting path can be identified, the delegated access can be controlled, and intrusions can be detected [26]. Similarly, the reply may also need to be signed by the delegates as it is sent back to the client. The delegated accesses over the Internet motivate the need for an efficient multisignature scheme. With digital multisignatures, all delegates can sign a mobile code in serial or

Manuscript received July 3, 1997; revised March 9, 1999. This work was supported by the National Science Council, Taiwan, under Contract NSC 89-2213-E-009-004.

The authors are with the Department of Computer Science and Information Engineering, National Chiao Tung University, Hsinchu 30010, Taiwan, R.O.C.

Publisher Item Identifier S 0018-9545(00)03524-6.

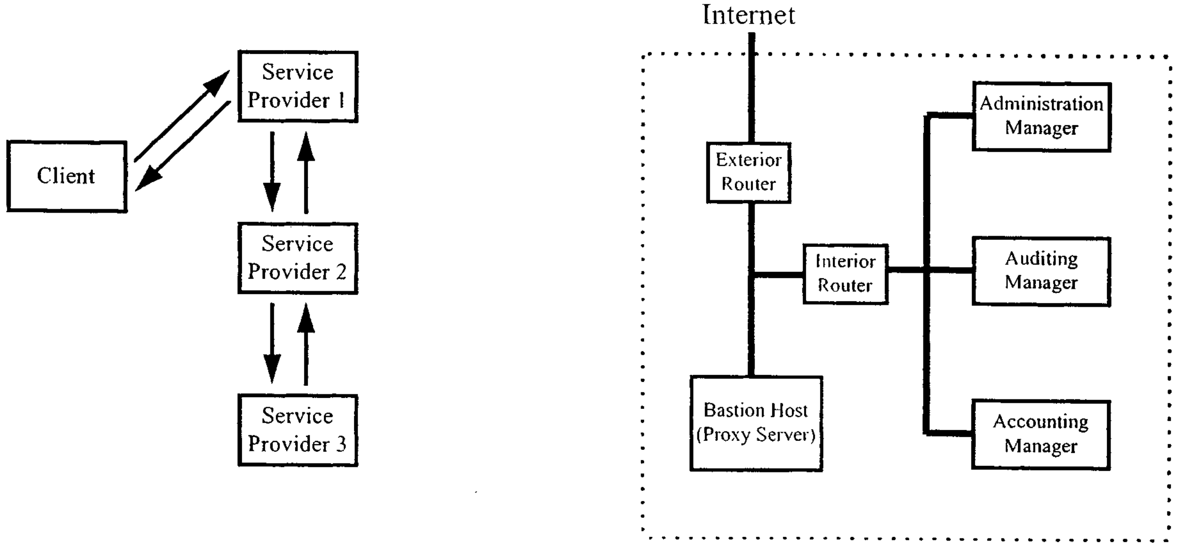


Fig. 1. (a) Delegated accesses. (b) Firewall structure of a service provider.

parallel, and consequently the receiver can identify the signers of the mobile code and determine the access privileges of the code.

Consider a typical example shown in Fig. 1. In Fig. 1(a), a service is partitioned among three service providers. In response to the client's request for service, each service provider only provides part of the service. Suppose a client issues a request and sends it along with its mobile code and data to the service provider SP_1. Upon receiving the request package, SP_1 provides part of the service and forward the request package to the next service provider SP_2. SP_2 repeats the same process. As the service providers forward the request package, they must sign the package serially so that the receiving server can verify the requesting path. Thus, an efficient serial multisignature scheme is needed. The service provider usually uses a firewall structure to protect itself [see Fig. 1(b)]. In the firewall structure, the proxy servers provide services to external clients on behalf of internal servers. When a request package arrives, the proxy server must acquire the permission from all system managers before it can process the request. If a system manager grants the request, it must sign the request and send it back to the proxy server. As the proxy server acquires the signatures from all system managers, it combines the signatures to a single multisignature. In this way, nonrepudiation can be achieved. In this context, we need an efficient parallel multisignature scheme.

There are two modes of digital signature schemes, the appendix mode and the message recovery mode. By the appendix mode, the digital signature is sent to a receiver along with the corresponding message. The message itself is not encrypted and will be used for verification by the receiver. The famous digital signature scheme using the appendix mode is the ElGamal digital signature scheme which is based on the discrete logarithm problem [3].

By the message recovery mode, the signed message is embedded in the signature, and can be recovered from the signature. The famous digital signature scheme with message recovery is the RSA digital signature scheme which is based on the difficulty of factoring large integers [22]. In 1993, Nyberg and Rueppel proposed the first discrete logarithm based scheme with message recovery [17]. Other digital signature schemes with message recovery based on the discrete logarithm problem are also proposed later [11], [23]. Some of these schemes have the capability of data encryption to guarantee the privacy of signed messages [12], [13], [15], [19], [20]. Thus, only the legal receiver can recover the original message from the signature and verify its authenticity. However, all these digital signature schemes only allow a single signer to sign a message.

If many signers want to sign the same message, the easy solution has each of them signing separately. In this way, total size of the signatures for a message is proportional to the number of signers. As the number of signers grows, total size of the signatures may become unacceptable large. Repudiation is also possible with this approach. Consider the following case. A signer u_i signs a message and forward the signature together with the message to the next signer u_{i+1}. u_{i+1} signs the message only if u_i has signed it. After all signers have signed the message, signer u_i simply removes his signature and claims that he has never signed the message. In this way, the signers, who signed the message after u_i, are cheated. Therefore, multisignature schemes are needed to resolve the problems.

There are many digital multisignature schemes proposed in the past based on RSA [2], [8], [9], [21]. However, with these schemes, either the signers and signing order must be determined in advance, or the size of a multisignature grows proportional to the number of signers. These multisignature schemes cannot be used to support mobile services because the users are mobile and the delegates cannot be predetermined. In 1994, Harn proposed a parallel digital multisignature scheme based on the discrete logarithm problem [10]. The scheme allows multiple signers to sign the same message separately and all individual signatures can be combined into a multisignature. In Harn's scheme, however, the messages needed to exchange among signers are heavy and an additional redundancy scheme is needed for verification. This makes the scheme difficult to use. Furthermore, Harn's scheme does not support message recovery and its messages are not encrypted.

In this paper, we first propose a new digital signature schemes with message recovery in Section II. Unlike conventional digital signature schemes, our scheme does not need any additional one-way hash functions or redundancy schemes to eliminate the possibility of forging the signature. Based on the signature scheme, a serial and a parallel digital multisignature schemes, which are suitable for open mobile networks, are proposed in Section III. The serial multisignature scheme allows all users to sign the message serially, that is, to sign the message one by one. Unlike those digital multisignature schemes based on RSA, our serial digital multisignature scheme does not need to predetermine the signing order. The parallel multisignature scheme allows each user to sign the same message separately and independently, and then all individual signatures can be combined into a multisignature. The parallel scheme needs fewer messages for generating the combined multisignatures. Note that the proposed schemes in Section III can not guarantee the privacy of signed messages. That is, an interceptor is also able to recover the original message from the signature with the public information. Thus, an additional procedure is proposed in the security analysis of Section IV to prevent the disclosure attack. In addition, other security analysis and performance comparison are given in this section.

II. Proposed New Digital Signature Scheme with Message Recovery

Let p be a large prime and α be a primitive element in GF(p). p and α are known by all users. Each user chooses its private key X_i uniformly between 0 and $p-1$ such that gcd$(X_i, p-1) = 1$ and computes $Y_i \equiv (\alpha)^{X_i} \bmod p$ as its public key. When any signer U_i wants to sign the message m, $m \in Z_p$, and sends the signature to the receiver U_j, U_i and U_j follow the following steps:

1) U_i—The Signature Generation:

Step 1) compute S, where

$$S \equiv (Y_i)^m \bmod p \tag{1}$$

Step 2) select a number k_i randomly between 1 and $p-1$ and compute

$$r_i \equiv [m \cdot (\alpha)^{-k_i}] \bmod p \tag{2}$$

Step 3) solve the congruence

$$(S + t_i) \equiv (X_i)^{-1}(k_i - r_i) \bmod (p-1) \tag{3}$$

for the integer t_i.

The signature for the message m signed by U_i is then the triple $\{t_i, r_i, S\}$.

2) U_j—The Signature Verification and Recovery: The receiver U_j follows the following steps to recover m and verify the authenticity of the initial signature.

Step 1) perform the following equation to recover m.

$$\begin{aligned}(Y_i)^{S+t_i} &\cdot r_i \cdot (\alpha)^{r_i}\\ &\equiv [(\alpha)^{X_i(S+t_i)} \cdot m \cdot (\alpha)^{-k_i} \cdot (\alpha)^{r_i}] \bmod p\\ &\equiv [(\alpha)^{k_i - r_i} \cdot m \cdot (\alpha)^{-k_i + r_i}] \bmod p\\ &\equiv m \bmod p \end{aligned}\tag{4}$$

Step 2) see whether m recovered by (4) satisfies (1)

$$S \equiv (Y_i)^m \bmod p$$

If the above equation holds, the authenticity of the initial signature is verified.

III. The Digital Multisignature Schemes

We will use the basic scheme presented in Section II to design two digital multisignature schemes: the parallel multisignature scheme and the serial multisignature scheme. To simplify the representation, the following notations are the same as those mentioned in Section II. The public information consists of p, α, the user's public key Y_i and the signature $\{t_i, r_i, S\}$. And the secret information included the user's private key X_i and random number k_i.

A. The Parallel Multisignature Scheme

The parallel multisignature scheme allows multiple signers to sign the same message separately and then combine all individual signatures into a multisignature. The message m is first signed by an initiator U_1, and then is sent separately to all signers. Finally, U_1 is responsible for combining these individual signatures into a multisignature. In the example of Fig. 1(b) the initiator U_1 is the proxy server of the firewall structure.

1) Generation of a Combined Multisignature:

a) Initial phase:

1) The generation procedure of the basic signature for the message m, $m \in Z_p$, by this variant parallel scheme is as follows:

Step 1) compute S, where

$$S \equiv (Y_1)^m \bmod p \tag{5}$$

Step 2) select a number k_1 randomly between 1 and $p-1$ and compute

$$r_1 \equiv [m \cdot (\alpha)^{-k_1}] \bmod p \tag{6}$$

Step 3) solve the congruence

$$(S + t_1) \equiv (X_1)^{-1}(k_1 - r_1) \bmod (p-1) \tag{7}$$

for the integer t_1.

The basic signature for the message m signed by U_1 is then the triple $\{t_1, r_1, S\}$.

After U_1 generates the basic signature, U_1 sends $(\{t_1, r_1, S\}, U_1)$ to all the other users $U_2, U_3, \ldots, U_n$. The number k_1 must be kept secret.

2) When any other signer U_j, $2 \le j \le n$, receives $(\{t_1, r_1, S\}, U_1)$, the following steps will be performed:

b) The basic signature verification: When any other signer $U_j, 2 \le j \le n$, receives $(\{t_1, r_1, S\}, U_1)$, U_j try to recover the message m and verify the authenticity of the basic signature $\{t_1, r_1, S\}$ by using the public key of U_1. U_j recovers m by performing

$$(Y_1)^{S+t_1} \cdot r_1 \cdot (\alpha)^{r_1} \equiv m \bmod p \tag{8}$$

If the message m recovered by (8) satisfies $S \equiv (Y_1)^m \bmod p$, U_j verifies the authenticity of the basic signature.

c) The individual signatures generation: If U_j agrees to sign m, U_j follows the following steps to sign m and generates its individual signature.

Step 1) select a number k_j randomly between 1 and $p-1$ and compute

$$r_j \equiv [m \cdot (\alpha)^{-k_j}] \bmod p \tag{9}$$

Step 2) solve the congruence

$$(S + t_j) \equiv (X_j)^{-1}(k_j - r_j) \bmod (p-1) \tag{10}$$

for the integer t_j.

The individual signature of U_j for the message m is then the triple $\{t_j, r_j, S\}$.

After U_j signs the message m and generates the individual signature for m, U_j sends $(\{t_j, r_j, S\}, U_j)$ back to the initiator U_1 and keeps the number k_j privately.

2) Generation Phase:

1) When U_1 receives any individual signature, $(\{t_j, r_j, S\}, U_j)$, $2 \leq j \leq n$:

a) The individual signatures verification: when U_1 receives any individual signature, $(\{t_j, r_j, S\}, U_j)$, $2 \leq j \leq n$, U_1 recovers the message m and verify the authenticity of the individual signature by performing:

$$(Y_j)^{S+t_j} \cdot r_j \cdot (\alpha)^{r_j} \equiv m \bmod p \tag{11}$$

After the message m is recovered by (11), U_1 compares the recovered message m with the original message he sent to U_j and determines whether the two messages are identical. If the two messages are identical, the authenticity of U_j's individual signature $\{t_j, r_j, S\}$ is verified.

When the individual signature is verified successfully (that is, the message recovered from the individual signature is identical to the original message), U_1 then recovers $(\alpha)^{-k_j} \bmod p$ from r_j by performing

$$\begin{aligned} r_j \cdot m^{-1} &\equiv [m \cdot (\alpha)^{-k_j} \cdot m^{-1}] \bmod p \\ &\equiv (\alpha)^{-k_j} \bmod p \end{aligned} \tag{12}$$

b) The combined multisignatures generation: Once U_1 receives all individual signatures and these individual signatures all pass the verification, all $(\alpha)^{-k_j}$ are computed. U_1 selects another number k randomly between 1 and $p-1$ such that $k \neq k_1$, and then computes R.

$$\begin{aligned} R \equiv [&m \cdot (\alpha)^{-k_2} \cdot \ldots \cdot (\alpha)^{-k_n} \cdot (\alpha)^{-k} \cdot (\alpha)^{r_2} \cdot \ldots \cdot (\alpha)^{r_n} \\ &\cdot (Y_2)^{t_2} \cdot \ldots \cdot (Y_n)^{t_n}] \bmod p \end{aligned} \tag{13}$$

And then U_1 solves the congruence

$$(S + T) \equiv (X_1)^{-1}(k - R) \bmod (p-1) \tag{14}$$

for the integer T.

As a result, the combined multisignature of U_1, $U_2, \cdots, U_{(n-1)}$ and U_n for the message m is $\{T, R, S\}$. U_1 also keeps the number k privately.

3) Verification of a Combined Multisignature:

1) After receiving the combined multisignature $\{T, R, S\}$ for the message m, an external receiver/verifier needs all signers' public key Y_i to verify the authenticity of the multisignature and to recover m from the multisignature. Then the external verifier follows the following steps to recover m and verify the authenticity of the combined multisignature.

Step 1) perform the following equation to recover m.

$$\begin{aligned} &Y_1^{S+T} \cdot Y_2^S \cdot \ldots \cdot Y_n^S \cdot R \cdot \alpha^R \\ &\quad \equiv [(\alpha^{X_1})^{S+T}] \cdot [(\alpha^{X_2})^S] \cdot \ldots \cdot [(\alpha^{X_n})S] \\ &\qquad \cdot [m \cdot \alpha^{-k_2} \cdot \ldots \cdot \alpha^{-k_n} \cdot \alpha^{-k} \cdot \alpha^{r_2} \cdot \ldots \\ &\qquad \ldots \cdot \alpha^{r_n} \cdot Y_2^{t_2} \cdot \ldots \cdot Y_n^{t_n}] \cdot \alpha^R \\ &\quad \equiv [(\alpha^{X_1})^{S+T}] \cdot [(\alpha^{X_2})^S] \cdot \ldots \cdot [(\alpha^{X_n})^S] \\ &\qquad \cdot [m \cdot \alpha^{-k_2} \cdot \ldots \cdot \alpha^{-k_n} \cdot \alpha^{-k} \cdot \alpha^{r_2} \cdot \ldots \\ &\qquad \ldots \cdot \alpha^{r_n} \cdot (\alpha^{X_2})^{t_2} \cdot \ldots \cdot (\alpha^{X_n})^{t_n}] \cdot \alpha^R \\ &\quad \equiv m \cdot [(\alpha^{X_1})^{S+T}] \cdot [(\alpha^{X_2})^{S+t_2}] \\ &\qquad \cdot \ldots \cdot [(\alpha^{X_n})^{S+t_n}] \cdot \alpha^{-k_2+r_2} \cdot \ldots \cdot \alpha^{-k_n+r_n} \\ &\qquad \cdot \alpha^{-k} \cdot \alpha^R \\ &\quad \equiv m \cdot [(\alpha^{X_1})^{(X_1^{-1})\,(k-R)}] \cdot [(\alpha^{X_2})^{(X_2^{-1})\,(k_2-r_2)}] \\ &\qquad \cdot \ldots \cdot [(\alpha^{X_n})^{(X_n^{-1})\,(k_n-r_n)}] \cdot \alpha^{-k_2+r_2} \\ &\qquad \cdot \ldots \cdot \alpha^{-k_n+r_n} \cdot \alpha^{-k+R} \\ &\quad \equiv m \bmod p \end{aligned} \tag{15}$$

Step 2) determine whether the message m recovered by (15) satisfies (5),

$$S \equiv (Y_1)^m \bmod p.$$

If the above equation holds, the authenticity of the combined multisignature is verified.

B. The Serial Multisignature Scheme

In many secure network systems, there is a center, e.g., the authentication server, which is trusted by all users. In the serial multisignature scheme, the trusted center can play the role of the public notary (PN). The responsibility of PN is to endorse the signatures and manage users' public keys. Without the trusted PN, some attacks may succeed (see Attack 5 in Section IV).

1) Generation of Multisignatures: Suppose that there are n users who need to sign a message, $m \in Z_p$. Without loss of generality, we assume that these n signers are $U_1, U_2, \ldots, U_{n-1}$ and U_n, and U_1 is the first signer while U_n is the last signer. The generation procedure of the multisignature for the message m is as follows:

1) **The initial signature generation by the first signer U_1:** When the first signer U_1 wants to sign the message m, U_1 follows the following steps.

Step 1) compute S, where

$$S \equiv (Y_1)^m \bmod p \tag{16}$$

Step 2) select a number k_1 randomly between 1 and $p-1$ and compute

$$r_1 \equiv [m \cdot (\alpha)^{-k_1}] \bmod p \tag{17}$$

Step 3) solve the congruence

$$(S+t_1) \equiv (X_1)^{-1}(k_1 - r_1) \bmod (p-1) \tag{18}$$

for the integer t_1.

The signature for the message m signed by U_1 is then the triple $\{t_1, r_1, S\}$. After U_1 signs the message m, U_1 sends $(\{t_1, r_1, S\}, U_1)$ to PN to endorse the signature, and keeps the number k_1 privately. U_1 in $(\{t_1, r_1, S\}, U_1)$ indicates that the signature has signed by the first signer.

2) **The initial signature verification and multisignature generation by the Public Notary.** PN endorses m which has been signed by U_1 and records that a new multisignature is initialized by U_1:

a) The initial signature verification: Upon receiving $(\{t_1, r_1, S\}, U_1)$, PN recovers m and verifies the authenticity of the initial signature $\{t_1, r_1, S\}$ by using U_1's public key Y_1 as below:

Step 1) perform the following equation to recover m.

$$\begin{aligned}(Y_1)^{S+t_1} \cdot r_1 \cdot (\alpha)^{r_1} &\equiv [(\alpha)^{X_1(S+t_1)} \cdot m \cdot (\alpha)^{-k_1} \cdot (\alpha)^{r_1}] \bmod p\\ &\equiv [(\alpha)^{k_1-r_1} \cdot m \cdot (\alpha)^{-k_1+r_1}] \bmod p\\ &\equiv m \bmod p\end{aligned} \tag{19}$$

Step 2) determine whether m recovered by (19) satisfies (16),

$$S \equiv (Y_1)^m \bmod p$$

If the above equation holds, the authenticity of the initial signature is verified.

b) The multisignature generation: If PN agrees to sign m, PN follows the following steps to sign m and generates the multisignature.

Step 1) select a number k_{pn} randomly between 1 and $p-1$ and compute

$$r_{pn} \equiv [(Y_1)^{t_1} \cdot r_1 \cdot (\alpha)^{r_1} \cdot (\alpha)^{-k_{pn}}] \bmod p \tag{20}$$

Step 2) solve the congruence

$$(S+t_{pn}) \equiv (X_{pn})^{-1}(k_{pn} - r_{pn}) \bmod (p-1) \tag{21}$$

for the integer t_{pn}.

The multisignature signed by U_1 and endorsed by the Public Notary PN for the message m is then $\{t_{pn}, r_{pn}, S\}$. After PN endorses the message m and generates the multisignature for m, PN sends $(\{t_{pn}, r_{pn}, S\}, U_1,$ PN) to next signer U_2 and keeps the number k_{pn} privately.

3) **The multisignature verification and the next multisignature generation by the signer U_2:**

c) The initial signature verification: Upon receiving $(\{t_{pn}, r_{pn}, S\}, U_1,$ PN), U_2 recovers m and verifies the authenticity of the multisignature $\{t_{pn}, r_{pn}, S\}$ by using the public keys of U_1 and PN. U_2 follows the following steps to recover m and verify the authenticity of the initial signature.

Step 1) perform the following equation to recover m.

$$\begin{aligned}(Y_{pn})^{S+t_{pn}} \cdot r_{pn} \cdot (\alpha)^{r_{pn}} \cdot Y_1^S &\equiv [\alpha^{X_{pn}(S+t_{pn})}] \cdot [(Y_1)^{t_1} \cdot r_1 \cdot (\alpha)^{r_1} \cdot (\alpha)^{-k_{pn}}]\\ &\quad \cdot a^{r_{pn}} \cdot Y_1^S\\ &\equiv [\alpha^{X_{pn}(S+t_{pn})}] \cdot [(\alpha^{X_1})^{t_1} \cdot m \cdot (\alpha)^{-k_1}\\ &\quad \cdot (\alpha)^{r_1} \cdot (\alpha)^{-k_{pn}}] \cdot a^{r_{pn}} \cdot (\alpha^{X_1})^S\\ &\equiv m \cdot \alpha^{X_{pn}(S+t_{pn})} \cdot \alpha^{X_1(S+t_1)} \cdot (\alpha)^{-k_{pn}+r_{pn}}\\ &\quad \cdot (\alpha)^{-k_1+r_1}\\ &\equiv m \cdot \alpha^{X_{pn}({X_{pn}}^{-1})(k_{pn}-r_{pn})} \cdot \alpha^{X_1(X_1^{-1})(k_1-r_1)}\\ &\quad \cdot (\alpha)^{-k_{pn}+r_{pn}} \cdot (\alpha)^{-k_1+r_1}\\ &\equiv m \bmod p\end{aligned} \tag{22}$$

Step 2) determine whether m recovered by (22) satisfies (16),

$$S \equiv (Y_1)^m \bmod p$$

If the above equation holds, the authenticity of the initial signature is verified.

d) The multisignature generation: If U_2 agrees to sign m, U_2 follows the following steps to sign m and generates the multisignature.

Step 1) select a number k_2 randomly between 1 and $p-1$ and compute

$$r_2 \equiv [r_{pn} \cdot (\alpha)^{-k_2}] \bmod p \tag{23}$$

Step 2) solve the congruence

$$(S+t_2) \equiv (X_2)^{-1}(k_2 - r_2) \bmod (p-1) \tag{24}$$

for the integer t_2.

The multisignature signed by U_1, PN and U_2 for the message m is then $\{t_{pn}, t_2, r_2, S\}$. After U_2 signs the message m and generates the multisignature for m, U_2 sends $(\{t_{pn}, t_2, r_2, S\}, U_1, U_2)$ to next signer U_3 and keeps the number k_2 privately.

4) **The multisignature verification and the next multisignature generation by the ith signer U_i, where 3 $\leq i \leq n$:**

e) The multisignature verification: When receiving $(\{t_{pn}, t_{i-1}, r_{i-1}, S\}, U_1, U_2, \ldots, U_{i-1})$, U_i will recover $m$and verify the authenticity of the multisignature $\{t_{pn}, t_{i-1}, r_{i-1}, S\}$ by using the public keys of the signers U_1, $U_2, \ldots, U_{i-1}$ and PN who have signed the message m before. U_i follows the following steps to recover m and verify the authenticity of the multisignature.

Step 1) perform the following equation to recover r_{pn}.

$$\begin{aligned}
(Y_2\cdots Y_{i-1})^S\cdot(Y_{i-1})^{t_{i-1}}\cdot r_{i-1}\cdot(\alpha)^{r_{i-1}}
&\equiv[(\alpha^{X_2})^S\cdot\ldots\cdot(\alpha^{X_{i-1}})^S]\cdot(\alpha^{X_{i-1}})^{t_{i-1}}\\
&\quad\cdot[r_{pn}\cdot(Y_2)^{t_2}\cdot\ldots\cdot(Y_{i-2})^{t_{i-2}}\\
&\quad\cdot(\alpha)^{-k_2+r_2-\cdots-k_{i-2}+r_{i-2}}\cdot(\alpha)^{-k_{i-1}}]\cdot(\alpha)^{r_{i-1}}\\
&\equiv r_{pn}\cdot[(\alpha^{X_2})^S\cdot\ldots\cdot(\alpha^{X_{i-1}})^S]\cdot[(\alpha^{X_2})^{t_2}\\
&\quad\cdot\ldots\cdot(\alpha^{X_{i-1}})^{t_{i-1}}]\cdot(\alpha)^{-k_2+r_2-\cdots-k_{i-1}+r_{i-1}}\\
&\equiv r_{pn}\cdot(\alpha^{X_2})^{S+t_2}\cdot\ldots\cdot(\alpha^{X_{i-1}})^{S+t_{i-1}}\\
&\quad\cdot(\alpha)^{-k_2+r_2-\cdots-k_{i-1}+r_{i-1}}\\
&\equiv r_{pn}\cdot(\alpha^{X_2})^{(X_2^{-1})\,(k_2-r_2)}\\
&\quad\cdot\ldots\cdot(\alpha^{X_{i-1}})^{(X_{i-1}^{-1})\,(k_{i-1}-r_{i-1})}\\
&\quad\cdot(\alpha)^{-k_2+r_2-\cdots-k_{i-1}+r_{i-1}}\\
&\equiv r_{pn}\bmod p \qquad (25)
\end{aligned}$$

(Refer to (27) for the expression of r_{i-1}.)

Step 2) perform (22) to recover m.

$$(Y_{pn})^{S+t_{pn}}\cdot r_{pn}\cdot(\alpha)^{r_{pn}}\cdot Y_1^S\equiv m\bmod p \qquad (26)$$

Step 3) determine whether the message m recovered by (26) satisfies (16),

$$S\equiv(Y_1)^m\bmod p$$

If the above equation holds, the authenticity of the initial signature is verified.

f) The next multisignature generation: If U_i agrees to sign m, U_i follows the following steps to sign m and generates the next multisignature.

Step 1) select a number k_i randomly between 1 and $p-1$, and compute

$$\begin{aligned}
r_i&\equiv[(Y_{i-1})^{t_{i-1}}\cdot r_{i-1}\cdot(\alpha)^{r_{i-1}}\cdot(\alpha)^{-k_i}]\bmod p\\
&\equiv[(Y_{i-1})^{t_{i-1}}\cdot(Y_{i-2})^{t_{i-2}}\cdot r_{i-2}\cdot(\alpha)^{r_{i-2}}\\
&\quad\cdot(\alpha)^{-k_{i-1}}\cdot(\alpha)^{r_{i-1}}\cdot(\alpha)^{-k_i}]\bmod p\\
&\equiv[(Y_{i-1})^{t_{i-1}}\cdot(Y_{i-2})^{t_{i-2}}\cdot\ldots\cdot(Y_2)^{t_2}\cdot r_2\\
&\quad\cdot(\alpha)^{r_2}\cdot(\alpha)^{-k_3}\cdot\ldots\cdot(\alpha)^{r_{i-2}}\cdot(\alpha)^{-k_{i-1}}\cdot(\alpha)^{r_{i-1}}\\
&\quad\cdot(\alpha)^{-k_i}]\bmod p\\
&\equiv[(Y_{i-1})^{t_{i-1}}\cdot(Y_{i-2})^{t_{i-2}}\cdot\ldots\cdot(Y_2)^{t_2}\cdot[r_{pn}\cdot(\alpha)^{-k_2}]\\
&\quad\cdot(\alpha)^{r_2}\cdot(\alpha)^{-k_3+r_3}\cdot\ldots\cdot(\alpha)^{-k_{i-1}+r_{i-1}}\\
&\quad\cdot(\alpha)^{-k_i}]\bmod p\\
&\equiv[r_{pn}\cdot(Y_2)^{t_2}\cdot\ldots\cdot(Y_{i-1})^{t_{i-1}}\\
&\quad\cdot(\alpha)^{-k_2+r_2-\cdots-k_{i-1}+r_{i-1}}\cdot(\alpha)^{-k_i}]\bmod p \qquad (27)
\end{aligned}$$

Step 2) solve the congruence

$$(S+t_i)\equiv(X_i)^{-1}(k_i-r_i)\bmod(p-1) \qquad (28)$$

for the integer t_i.

After U_i signs the message m and generates the next multisignature for m, U_i sends $(\{t_{pn},t_i,r_i,S\}$, U_1, $U_2,\cdots,U_i)$ to the next signer $U_{(i+1)}$ and keeps the number k_i privately. If $i=n$, then the multisignature is $\{t_{pn},t_n,r_n,S\}$ which has been signed by U_1, $U_2,\cdots,U_{(n-1)}$, U_n and endorsed by PN for the message m.

2) Verification of Multisignatures: After receiving the final multisignature $\{t_{pn},t_n,r_n,S\}$, for the message m, any external receiver/verifier needs to use all signers' public keys to verify the authenticity of the multisignature and recover m from the multisignature. The external receiver follows the following steps to recover m and verify the authenticity of the multisignature.

Step 1) perform the following equation to recover r_{pn}.

$$\begin{aligned}
(Y_2\cdots Y_n)^S\cdot(Y_n)^{t_n}\cdot r_n\cdot(\alpha)^{r_n}
&\equiv[(\alpha^{X_2})^S\cdot\ldots\cdot(\alpha^{X_n})^S]\cdot(\alpha^{X_n})^{t_n}\\
&\quad\cdot[r_{pn}\cdot(Y_2)^{t_2}\cdot\ldots\cdot(Y_{n-1})^{t_{n-1}}\\
&\quad\cdot(\alpha)^{-k_2+r_2-\cdots-k_{n-1}+r_{n-1}}\cdot(\alpha)^{-k_n}]\cdot(\alpha)^{r_n}\\
&\equiv r_{pn}\cdot(\alpha^{X_2})^S\cdot\ldots\cdot(\alpha^{X_n})^S\cdot(\alpha^{X_n})^{t_n}\\
&\quad\cdot(\alpha^{X_2})^{t_2}\cdot\ldots\cdot(\alpha^{X_{n-1}})^{t_{n-1}}\\
&\quad\cdot(\alpha)^{-k_2+r_2-\cdots-k_{n-1}+r_{n-1}}\cdot(\alpha)^{-k_n}]\cdot(\alpha)^{r_n}\\
&\equiv r_{pn}\cdot(\alpha^{X_2})^{S+t_2}\cdot\ldots\cdot(\alpha^{X_n})^{S+t_n}\\
&\quad\cdot(\alpha)^{-k_2+r_2-\cdots-k_n+r_n}\\
&\equiv r_{pn}\cdot(\alpha^{X_2})^{(X_2^{-1})\,(k_2-r_2)}\cdot\ldots\cdot(\alpha^{X_n})^{(X_n^{-1})\,(k_n-r_n)}\\
&\quad\cdot(\alpha)^{-k_2+r_2-\cdots-k_n+r_n}\\
&\equiv r_{pn}\bmod p \qquad (29)
\end{aligned}$$

Step 2) perform (22) to recover m.

$$(Y_{pn})^{S+t_{pn}}\cdot r_{pn}\cdot(\alpha)^{r_{pn}}\cdot Y_1^S\equiv m\bmod p \qquad (30)$$

Step 3) see whether m recovered by (30) satisfies (16),

$$S\equiv(Y_1)^m\bmod p.$$

If the above equation holds, the authenticity of the multisignature is verified.

IV. The Security Analysis and Performance Comparisons

In Section IV-A, we will analyze the security of the proposed basic scheme and the two expanded digital multisignature schemes. The performance of the two multisignature schemes will be compared with other known multisignature schemes in Section IV-B.

A. The Security Analysis

In this section, we analyze the possible attacks against the digital signature and multisignature schemes with message recovery. Like the types of attacks described in [3], the attacks to our schemes can be divided into three types. The first type is to get private keys of the users. The second type of attacks is to forge the signature $\{t_i, r_i, S\}$, any multisignatures, $\{T, R, S\}$ or $\{t_{pn}, t_j, r_j, S\}$, where $2 \leq j \leq n$. In addition, for the applications that must ensure the secrecy of signed messages, the third type of attacks is to disclose the message from the signature.

1) Attacks Aiming to Get Private Keys:

a) Attack 1: Get the private key of a signer.

There are three possible approaches to get the private keys of a signer.

1) Recover X_i directly from Y_i: Since $Y_i \equiv (\alpha)^{X_i} \bmod p$, to recover private key X_i of user U_i from the corresponding public key Y_i is equivalent to solving the discrete logarithm problem.
2) Determine X_i from the set of signatures generated by U_i: By collecting a set of signatures generated by U_i, $(\{t_{i1}, r_{i1}, S_1\}, \{t_{i2}, r_{i2}, S_2\}, \cdots, \{t_{iw}, r_{iw}, S_w\})$ for w different messages, an intruder may try to solve the w equations of the form $(S_j+t_{ij}) \equiv (X_i)^{-1}(k_{ij}-r_{ij}) \bmod (p-1)$, where $1 \leq j \leq w$. Since there are $w+1$ unknowns (since each multisignature uses different secret k_{ij}), the system of equations is underdetermined and the private key of U_i is secure.
3) Recover any secret k_{ij} and then determine X_i by k_{ij}: An intruder may try to recover some k_{ij} directly from r_{ij} or to determine k_{ij} by solving the system of equations, $(S_j + t_{ij}) \equiv (X_i)^{-1}(k_{ij} - r_{ij}) \bmod (p-1)$ mentioned above. If an intruder can get some secret number k_{ij}, it can determine X_i by solving the equation $(S_j + t_{ij}) \equiv (X_i)^{-1}(k_{ij} - r_{ij}) \bmod (p-1)$. Although anyone may be able to collect a set of signatures generated by U_i, it is unworkable to compute any secret number k_{ij} with the system of equations because the number of unknowns is larger than the number of equations. On the other hand, to recover the value k_{ij} from $(\alpha)^{-k_{ij}} \bmod p$ is equivalent to solving the discrete logarithm problem.

b) Attack 2: When the private keys of one or more users are lost or a group of legal users conspires, will the private keys of other users be exposed? Suppose that a group of legal users $(U_{L1}, U_{L2}, \cdots, U_{Lt})$ where $2 \leq t \leq n-1$ conspires, or their private keys have leaked. Under this condition, the private keys X_{Lj} and some secret numbers k_{Lj}, where $1 \leq j \leq t$, of these users are not secure anymore. When the conspirators or intruders who holds these secret information intend to get the private keys of the other users who are still trusted or secure, the ways which they can use to break the security of the other secure users are only the same approaches mentioned in attack 1. Therefore, even if the private keys of one or more users leak or a group of legal users conspires, the security of any other users will not be broken.

2) Attacks for Forging Multisignatures:

a) Attack 3: The substitution attack: There is a stronger form of forgery described in [3], [18]–[20], where a forger who knows a message m with the corresponding signature can generate some valid signatures for the messages of a special form $M \equiv m \cdot (\alpha)^e \bmod p$. Even though the resulting value M is uncontrolled, the stronger form of forgery attack is still dangerous for all ElGamal-type schemes and RSA. The attack is typically prevented by the use of a one-way hash function or a redundancy scheme.

But our serial digital multisignature scheme with message recovery can prevent the substitution attack without any additional one-way hash functions and redundancy schemes. That is, given the final multisignature $\{t_{pn}, t_n, r_n, S\}$ signed by $U_1, \cdots$, and U_n with the corresponding message m, the substitution attack is prevented as follows.

Let the forged multisignature be $\{t'_{pn}, t'_n, r'_n, S'\}$. A forger first selects e for the message

$$M = m \cdot \alpha^e \bmod p. \tag{31}$$

To guarantee that Step 3) of verification (i.e., $S \equiv (Y_1)^M \bmod p$) is successful, S' must be

$$(Y_1)^M \bmod p.$$

Then, to guarantee that the result of Step 2) of verification [i.e., (30)] is equal to M, the forger must determine t'_{pn} and r'_{pn} to satisfy the following equation:

$$(Y_{pn})^{S'+t'_{pn}} \cdot r'_{pn} \cdot (\alpha)^{r'_{pn}} \cdot Y_1^{S'} \equiv M \bmod p.$$

Based on (30) and (31), the above equation will be

$$\begin{aligned}(Y_{pn})^{S'+t'_{pn}} &\cdot r'_{pn} \cdot (\alpha)^{r'_{pn}} \cdot Y_1^{S'} \\ &\equiv M \bmod p \\ &\equiv m \cdot \alpha^e \bmod p \\ &\equiv (Y_{pn})^{S+t_{pn}} \cdot r_{pn} \cdot (\alpha)^{r_{pn}} \cdot Y_1^S \cdot \alpha^e \bmod p.\end{aligned} \tag{32}$$

In (32), only t'_{pn} and r'_{pn} are undetermined, but r'_{pn} is dependent on the result of Step 1) of verification [i.e., (29)]. That is,

$$(Y_2 \cdot Y_n)^{S'} \cdot (Y_n)^{t'_n} \cdot r'_n \cdot (\alpha)^{r'_n} \equiv r'_{pn} \bmod p.$$

No matter what value r'_{pn} is, (32) can be simplified as follows:

$$\begin{aligned}(Y_{pn})^{S'+t'_{pn}} \equiv (Y_{pn})^{S+t_{pn}} &\cdot r_{pn} \cdot (r'_{pn})^{-1} \cdot (\alpha)^{r_{pn}} \\ &\cdot ((\alpha)^{r'_{pn}})^{-1} \cdot Y_1^{S-S'} \cdot \alpha^e \bmod p.\end{aligned}$$

Obviously, determining t'_{pn} is equivalent to solving the discrete logarithm problem. Therefore, it is impossible to forge a multisignature with the message $M \equiv m \cdot (\alpha)^e \bmod p$. And our schemes can prevent the substitution attack without any additional one-way hash functions or redundancy schemes.

b) Attack 4: A forger may intend to forge the parallel signature $\{t_i, r_i, S\}$, the final multisignatures $\{T, R, S\}$ or the serial signature $\{t_{pn}, t_j, r_j, S\}$, for any given message M with only the public information.

Since the expressions of these signatures are similar, we give only an example to explain why the attack will fail. If a forger wants to forge the parallel signature $\{t'_i, r'_i, S'\}$, where $1 \le i \le n$, for any given message M, there are two possible approaches.

The first approach:

The forger first fixes M and t'_i and computes S' which satisfies $S' \equiv (Y_1)^M \bmod p$. Then the forger must compute r'_i to satisfy the following equation:

$$(Y_i)^{S'+t'_i} \cdot r'_i \cdot (\alpha)^{r'_i} \equiv M \bmod p.$$

Obviously, the computation is equivalent to solving the discrete logarithm problem.

The second approach:

The forger first fixes M and r'_i and computes S' which satisfies $S' \equiv (Y_1)^M \bmod p$. Then the forger must compute t'_i to satisfy the following equation:

$$(Y_i)^{S'+t'_i} \cdot r'_i \cdot (\alpha)^{r'_i} \equiv M \bmod p.$$

For the same reason in the first approach, it is also equivalent to solving the discrete logarithm problem.

c) Attack 5: In our serial digital multisignature scheme, a legal signer U_i, where $2 \le i \le n$, may want to forge a multisignature $\{t'_{pn}, t'_i, r'_i, S'\}$ for any given message M and then declares that M has signed by $U_1, U_2, \cdots, U_{i-1}$ and itself:

For any given message M, a legal user U_i is unable to forge the multisignature $\{t'_{pn}, t'_i, r'_i, S'\}$, where $2 \le i \le n$. Details are described as follows:

If U_i computes an S' which satisfies $S' \equiv (Y_1)^M \bmod p$, and selects three number t'_{pn}, t'_i and r'_i, then U_i will get r'_{pn}, where

$$(Y_2 \cdots Y_i)^{S'} \cdot (Y_i)^{t'_i} \cdot r'_i \cdot (\alpha)^{r'_i} \equiv r'_{pn} \bmod p$$

But U_i will find that

$$(Y_{pn})^{S'+t'_{pn}} \cdot r'_{pn} \cdot (\alpha)^{r'_{pn}} \cdot Y_1^{S'} \equiv Z \bmod p$$

for some message Z. Obviously, to find the corresponding t'_{pn}, t'_i and r'_i such that $Z = M$ is equivalent to solving the discrete logarithm problem. Thus U_i cannot forge a multisignature for message M without others' signing.

However, without the trusted PN, this attack may succeed. For example, if PN and U_i conspire, U_i can computs a Y'_{pn} which statisfies

$$(Y'_{pn})^{S'+t'_{pn}} \cdot r'_{pn} \cdot (\alpha)^{r'_{pn}} \cdot Y_1^{S'} \equiv M \bmod p, \text{ and}$$

PN announces that Y'_{pn} is his public key. The next user U_{i+1} will be cheated by the forged multisignature $\{t'_{pn}, t'_i, r'_i, S'\}$ and believe $U_1, U_2, \ldots,$ and U_i have signed M.

3) Attacks for Disclosing Messages:

a) Attack 6: The disclosure of signed messages: The proposed multisignature schemes in Sections II and III cannot guarantee the privacy of signed messages. For example, in the serial multisignature scheme, anyone who intercepts U_{i-1}'s signature $(\{t_{pn}, t_{i-1}, r_{i-1}, S\}, U_1, U_2 \ldots U_{i-1})$ can recover the original message m by following the steps of (25) and (26) with the public information, p, α, $Y_1 \ldots$ and Y_{i-1}.

The reason of proposing the schemes without the capability of data protection is that many applications with digital signatures allow the disclosure of messages. If the privacy of signed messages is critical, fortunately, only an additional encryption process is needed to enhance our proposed schemes.

The encryption process is straightforward. In brief, the sender U_i encrypts his signature element r_i with the receiver U_j's public information Y_j:

$$C_i \equiv Y_j^{-k_i+r_i} \cdot r_i \bmod p.$$

That is, the original signature $\{t_i, r_i, S\}$ is replaced by $\{t_i, C_i, S\}$. When U_j receives the signature, he follows the following steps to recover r_i.

Step 1) Compute

$$\begin{aligned} Y_i^S \cdot Y_i^{t_i} \bmod p &\equiv (\alpha^{X_i})^S \cdot (\alpha^{X_i})^{t_i} \bmod p \\ &\equiv \alpha^{X_i(S+t_i)} \bmod p \\ &\equiv \alpha^{X_i[X_i^{-1}(k_i-r_i)]} \bmod p \\ &\equiv \alpha^{(k_i-r_i)} \bmod p. \end{aligned}$$

Step 2) Compute

$$\begin{aligned} \alpha^{(k_i-r_i)X_j} \bmod p &\equiv (\alpha^{X_j})^{k_i-r_i} \bmod p \\ &\equiv Y_j^{k_i-r_i} \bmod p. \end{aligned}$$

Step 3) Recover r_i from C_i by performing

$$\begin{aligned} Y_j^{k_i-r_i} \cdot C_i \bmod p &\equiv Y_j^{k_i-r_i} \cdot (Y_j^{-k_i+r_i} \cdot r_i) \bmod p \\ &\equiv r_i \bmod p. \end{aligned}$$

Thus, U_j can get the original signature $\{t_i, r_i, S\}$ and follow the schemes in Section III to verify the signature. Since only U_j has the private key X_j, an interceptor cannot decrypt r_i by the above three steps. Consequently, the signed message m remains secret.

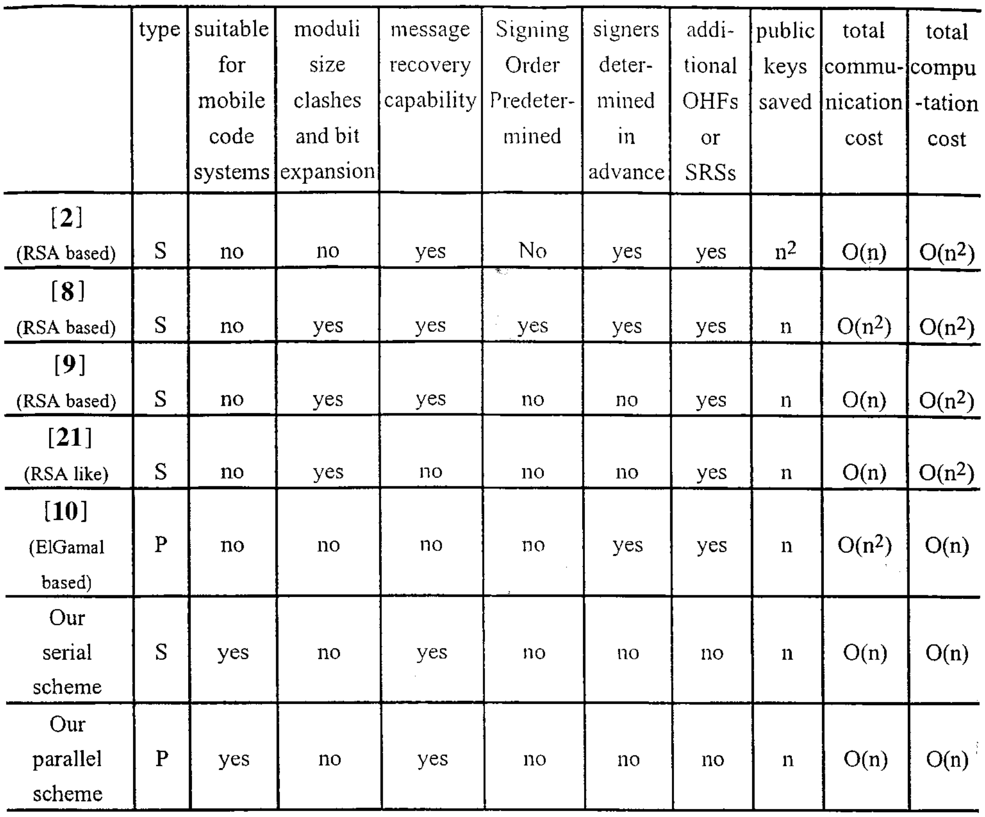
4) Discussion: Similar to the restriction of the ElGamal digital signature scheme where the secret number k cannot be used twice, our schemes have the restriction that the privately number k_i selected by U_i, where $1 \le i \le n$, cannot be used for more than once for different messages m and m'. If any k_i is used twice to generate the signatures/multisignatures, $\{t_i, r_i, S\}$ and $\{t'_i, r'_i, S'\}$, for m and m', respectively, then an intruder can derive the private key X_i of user U_i by solving

$$\begin{aligned} &[X_i(S+t_i)+r_i] \bmod (p-1) \\ &\quad \equiv k_i \equiv [X_i(S'+t'_i)+r'_i] \bmod (p-1) \end{aligned}$$

B. Comparisons

In this section, we will compare the performance of our schemes with other well-known schemes [2], [8], [9], [10], [21]. A number of RSA-based multisignature schemes have

TABLE I
THE COMPARISONS



	type	suitable for mobile code systems	moduli size clashes and bit expansion	message recovery capability	Signing Order Predetermined	signers determined in advance	additional OHFs or SRSs	public keys saved	total communication cost	total computation cost
[2] (RSA based)	S	no	no	yes	No	yes	yes	n2	O(n)	O(n2)
[8] (RSA based)	S	no	yes	yes	yes	yes	yes	n	O(n2)	O(n2)
[9] (RSA based)	S	no	yes	yes	no	no	yes	n	O(n)	O(n2)
[21] (RSA like)	S	no	yes	no	no	no	yes	n	O(n)	O(n2)
[10] (ElGamal based)	P	no	no	no	no	yes	yes	n	O(n2)	O(n)
Our serial scheme	S	yes	no	yes	no	no	no	n	O(n)	O(n)
Our parallel scheme	P	yes	no	yes	no	no	no	n	O(n)	O(n)

been proposed. The RSA-based multisignature schemes [8], [9], [21] have the drawbacks of bit expansion and moduli size clashes for multisignatures. In these schemes, the signing order of signers must be predetermined so that successive transformations on intermediate signatures can be applied. The latter signers are required to have a larger modulus than the former signers. This results in the expansion of signature size. These drawbacks make these schemes difficult to use in open mobile networks, where the number of mobile signers and the signing orders cannot be predetermined.

Harn's parallel scheme [10] can only be applied in a predetermined group of signers. In Harn's parallel scheme, each signer first broadcasts to all members of a group his own public, fresh signing information for a message. After receiving all other signers' signing information, a signer verifies the information, generates his own signature, and sends it to a designated clerk. A clerk, who takes the responsibility for collecting and verifying each individual signature, will produce a combined multisignature. Thus, n^2 messages are needed for the generation of a multisignature. Broadcasting and verifying signers' signing information is very costly and time-consuming. Furthermore, the scheme does not have the capability of message recovery, and therefore cannot assure the privacy of message contents.

Chang's scheme [2] does not resolve the problems of bit expansion and moduli size clashes. It requires that each signer of a closed group has n pairs of keys, and each recipient of the group needs to maintain n^2 key pairs for the verification. With the n key pairs, the signer can choose the appropriate key to sign an intermediate signature. In this way, the signing order need not be predetermined. However, the signer must be a member of the predetermined group. The management of the key pairs wastes precious storage and computation power of mobile systems. Chang's scheme is not scalable, and cannot be applied in open mobile networks, where the number of signers is large and cannot be predetermined.

Our schemes resolve the problems described above. The comparisons are summarized in Table I. The leftmost column lists the well-known schemes and each entry of the table indicates the property of a scheme with respect to an evaluation criterion. The first four schemes in Table I are RSA-based schemes and the last three schemes (include our schemes) are ElGamal-based. The terms used in the table are defined as follows:

OHF	One-way hash hunction.
S	Serial digital multisignature scheme.
M	Message recovery.
SRS	Suitable redundancy scheme.
P	Parallel digital multisignature scheme.
n	number of signers.

As shown in Table I, our schemes have the following properties. Our schemes are feasible in mobile code systems, and have the capability of message recovery, which ensures the privacy of message contents. The signers who will involve in signing do not need to be determined in advance, and the signing order of the serial scheme does not need to be predetermined. Our schemes can withstand the substitution attack (e.g., attack 4) without the need of any additional one-way hash functions and redundancy schemes. The size of the signature transmitted will not be expanded with respect to the number of signers. The total communication costs are low, and the total number of exponentiation operations in our schemes is low, compared with others.

V. Conclusion

In this paper, we propose a new digital signature scheme with message recovery based on the ElGamal scheme. The capability of message recovery has many advantages. Our basic scheme maintains the same security level of the original ElGamal scheme, but does not need any additional one-way hash functions or redundancy schemes to prevent the forgery of the signatures of some uncontrolled messages. Based on the basic digital signature scheme, a parallel multisignature scheme and a serial multisignature scheme are developed. The parallel digital multisignature scheme allows each user to sign the same message separately and independently. The serial digital multisignature scheme allows users to sign the message serially, but does not need to predetermine the signing order. The two digital multisignature schemes only need low computation and communication cost.

References

[1] T. Berners-Lee, R. Calilliau, A. Luotonen, H. F. Nielsen, and A. Secret, “The World Wide Web,” *Commun. ACM*, vol. 37, no. 8, pp. 77–82, Aug. 1994.

[2] C. C. Chang, E.-H. Lu, S.-F. Pon, and J.-Y. Lee, “Applying Harn-Kiesler multisignature scheme to electronic document systems,” in *Proc. National Information Security Conf.*, R.O.C., May 1995, pp. 35–38.

[3] T. ElGamal, “A public key cryptosystem and a signature scheme based on discrete logarithm,” *IEEE Trans. Inform. Theory*, vol. IT-31, pp. 469–472, July 1985.

[4] D. Dean, E. W. Felten, and D. S. Wallach, “Java security: From HotJava to Netscape and beyond,” in *Proc. IEEE Symp. Research in Security and Privacy*, May 1996.

[5] D. Flanagam, *Java in a Nutshell*: O’Reilly & Associates, Inc., Feb. 1996.

[6] J Gosling and H McGilton, “The Java Language Overview: A White Paper,”, Sun Microsystems Technical Report, May 1995.

[7] ——, *The Java Language Environment*: Sun Microsystems, May 1996.

[8] L. Harn and T. Kiesler, “New scheme for digital multisignature,” *Electron. Lett.*, vol. 25, no. 15, pp. 1002–1003, July 1989.

[9] ——, “RSA blocking and multisignature schemes with no bit expansion,” *Electron. Lett.*, vol. 26, no. 18, pp. 1490–1491, Aug. 1990.

[10] L. Harn, “New digital signature scheme based on discrete logarithm,” *Electron. Lett.*, vol. 30, no. 5, pp. 296–298, Mar. 1994.

[11] P. Horster, M. Michels, and H. Petersen, “Meta-message recovery and meta-blind signature schemes based on the discrete logarithm problem and their applications,” in *ASIACRYPT*, Australia: NSW, Dec. 1994, pp. 224–237.

[12] ——, “Authenticated encryption schemes with low communication costs,” *Electron. Lett.*, vol. 30, no. 15, pp. 1212–1213, July 1994.

[13] S. J. Hwang, C. C. Chang, and W. P. Yang, “An encryption signature scheme with low message expansion,” *J. Chinese Institute of Engineers*, vol. 18, no. 4, pp. 591–595, 1995.

[14] Security in JDK1.1: The JavaSecurity API and Digital Signatures, in Java Security Overview and Reference, Jan. 10, 1997.

[15] W. B. Lee and C. C. Chang, “Authenticated encryption scheme without using a one way function,” *Electron. Lett.*, vol. 31, no. 19, pp. 1656–1657, Sept. 1995.

[16] NCSA httpd Development Team. (1995, July) NCSA httpd. [Online] Available: http://hoohoo.ncsa.uiuc.edu/docs/Overview.html

[17] K. Nyberg and R. A. Rueppel, “A new signature scheme based on the DSA giving message recovery,” in *1st ACM Conference on Computer & Communications Security*, Fairfax, USA, Nov. 1993.

[18] K. Nyberg, “Comment: New digital signature scheme based on discrete logarithm,” *Electron. Lett.*, vol. 30, no. 6, p. 481, Mar. 1994.

[19] K. Nyberg and R. A. Rueppel, “Message recovery for signature schemes based on the discrete logarithm problem,” in *EURO-CRYPT’94* Perugia, Italy, May 1994, pp. 182–193.

[20] ——, “Message recovery for signature schemes based the on the discrete logarithm problem,” *Designs, Codes and Cryptography*, vol. 7, pp. 61–81, 1996.

[21] T. Okamoto, “A digital multisignature scheme using bijective public-key cryptosystems,” *ACM Trans. Comput. Syst.*, vol. 6, no. 8, pp. 432–441, Nov. 1988.

[22] M. Rivest, A. Shamir, and L. Adleman, “A method for obtaining digital signature and public-key cryptosystems,” *ACM Commun.*, vol. 21, pp. 120–126, 1978.

[23] J. M. Piveteau, “New signature scheme with message recovery,” *Electron. Lett.*, vol. 29, no. 25, p. 2185, Dec. 1993.

[24] P. Samarati, E. Bertino, and S. Jajodia, “An authorization model for a distributed hypertext system,” *IEEE Trans. Knowledge Data Eng.*, vol. 8, pp. 555–562, Aug. 1996.

[25] S.-P. Shieh and W.-H. Yang, “An authentication and key distribution protocol for open network systems,” *ACM Operating Systems Review*, pp. 32–41, Apr. 1996.

[26] S.-P. Shieh and V. D. Gligor, “On a pattern-oriented intrusion detection model,” IEEE Trans. Knowledge Data Eng., vol. 9, no. 4, pp. 661–668, Aug. 1997, to be published.

[27] S. P. Shieh, W. H. Yang, and H. M. Sun, “An authentication protocol without trusted third party,” *IEEE Commun. Lett.*, Apr. 1997.

[28] Sun Microsystems. (1996) “HotJava(TM): The Security Story”. [Online] Available: http://java.sun.com/1.0alpha3/doc/security/security.html

[29] F. Fellin, “Low level security in Java,” in *4th Int. World Wide Web Conf.*. Boston, MA, Dec. 1995, http://www.w3.org/pub/Conferences/WWW4/Papers/40.html.

Shiuh-Pyng Shieh (S’85–M’91) received the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Maryland, College Park, in 1986 and 1991, respectively.

He is currently the Director of Computer and Network Center, and a Professor with the Department of Computer Science and Information Engineering, National Chiao Tung University, Hsinchu, Taiwan, R.O.C. From 1988 to 1991, he participated in the design and implementation of the B2 Secure XENIX for IBM, Federal Sector Division, Gaithersburg, MD. He is also the designer of SNP (Secure Network Protocol). Since 1994, he has been a consultant for Computer and Communications Laboratory, Industrial Technology Research Institute, Taiwan. He is also a consultant for the National Security Bureau, Taiwan. His research interests include internetworking, distributed systems, and network security.

Dr. Shieh was on the organizing committees of a number of conferences, such as International Computer Symposium, and International conference on Parallel and Distributed Systems. Recently, he has been the General Chair of 1998 Network Security Technology Workshop, the Program Chair of 1999 Mobile Computing Conference, and 1997 Information Security Conference (INFOSEC’97).

Chern-Tang Lin, photograph and biography not available at the time of publication.

Wei-Bon Yang, photograph and biography not available at the time of publication.

Hung-Min Jun, photograph and biography not available at the time of publication.