JOURNAL OF KOREA MULTIMEDIA SOCIETY, VOL. 8, NO. 12, DECEMBER 2005(pp. 1638-1648)

Practical Revision for Implementing the Distributing Security-Mediated PKI

Jong-Phil Yang[†], Mi-Sun Shim[††], Kyung Hyune Rhee[†††]

ABSTRACT

The SEM approach to PKI offers several advantages, such as immediate revocation of users' signing ability without CRLs and compatibility with the standard RSA. However, it has a weakness against denial of service attack caused by breaking down or being compromised. G. Vanrenen et al. proposed a distributed SEM approach to overcome the weaknesses. However, it does not provide the desirable properties such as instant availability and immunity against denial of service attack, due to inadequate usage of threshold cryptography and proactive secret sharing. In this paper, we point out its structural contradictions and propose a modified version of distributed SEM approach.

Keywords: Peer to Peer, Certificate Revocation, Public Key Infrastructure, Secret Sharing

1. INTRODUCTION

Without doubt, the promise of public key infrastructure(PKI) technology has attracted a significant amount of attention in these days. The IETF PKIX Working Group is developing the Internet standards to support an X.509-based PKI. In PKI, a certificate is a signed binding a public key to certain properties. The correctness of the trust decisions which a relying party makes depends on the assumption that the entity knowing the matching private key possesses those properties. When this binding ceases to hold, this certificate needs to be revoked, and this revocation event must be propagated to relying parties, lest they make incorrect trust judgments regarding that public key. There are well-known mechanisms to solve the revocation of the certificate: Certificate Revocation List(CRL), Online Certificate Status Protocol(OCSP), delta CRL, indirect CRL, Certificate Revocation Tree(CRT) and Certificate Revocation System(CRS)[3,9].

In [4,19], Boneh et al. proposed an approach to fast certificate revocation centered around the concept of an on-line semi-trusted mediator(SEM). The SEM approach to PKI offers several advantages such as immediate revocation of users' signing ability without CRLs and compatibility with the standard RSA. However, it has a weakness against denial of service attack caused by breaking down or being compromised. To overcome the weakness, G. Vanrenen et al. proposed a distributed SEM approach[5]. However, in contrast to their opinions, it does not provide the desirable properties such as instant availability and immunity for denial of service attack, because of inadequate usage of threshold cryptography and proactive secret sharing. In this paper, we point out its structural contradictions and propose a modified version of

※ Corresponding Author : Kyung Hyune Rhee, Address : (608-737) 599-1, Daeyeon-3 Dong, Nam-Ku, Busan, Korea, TEL : +82-51-620-6395, FAX : +82-51-626-4887, E-mail : khrhee@pknu.ac.kr

Receipt date : Aug. 29, 2005, Approval date : Dec. 9, 2005

[†] Graduate School of Information Science and Electrical Engineering, Kyushu University
(E-mail : bogus@itslab.csce.kyushu-u.ac.jp)
[††] Infosec Technologies Co., Ltd.
(E-mail : ssssblue@infosec.co.kr)
[†††] Division of Electronic, Computer & Telecommunication Engineering, Pukyong National University

※ This research is a product of Information and Telecommunication National Scholarship Program supported by Ministry of Information and Communication (MIC) in Republic of Korea.

distributed SEM approach.

This paper is organized as follows. Section 2 reviews the original SEM approach and the distributed SEM approach. We discuss notable problems of the distributed SEM and present requirements for designing a modified version in Section 3. We present a modified version of the distributed SEM in Section 4. Section 5 discusses the security and the desirable features of our modified version. We conclude in Section 6.

2. RELATED WORK

2.1 SEM : Semi-trusted mediator

In [4,19], the SEM system is based on a variant of RSA which is called as mediated RSA(mRSA). As in RSA, each user has a public key (e, N) and a corresponding private key (d, N), where the modulus N is product of two large prime p and q, $\gcd(e, \phi(N)) = 1$ and $d \cdot e = 1 \bmod \phi(N)$. The public key of a user is the same as in the standard RSA. However, the two parts of a user's private key are d_{sem} and d_{user}, where $d = d_{sem} + d_{user}$ mod $\phi(N)$ d_{user} is the part held by the user and d_{sem} is the part held by the SEM. Since SEM must not know d_{user} and the user must not know d_{sem}, it is necessary to change RSA key setup procedure. That is, a CA(Certification Authority) generates the private key d instead of the user, chooses a random integer d_{sem} in $[1, N]$, and computes the last value as $d_{user} = d - d_{sem} \bmod \phi(N)$. Because the private key d is split into two halves, private key operations require the participation of both the user and SEM; each party raises the message to its half-exponent and the results are then multiplied.

The SEM approach provides several advantages such as compatibility with the standard RSA, immediate revocation of users' signing ability and no need for certificate revocation lists.

2.2 Distributing Security-Mediated PKI

In [5], G. Vanrenen et al. introduced disadvantages concerned with scalability. In their opinions, SEM approach has limitation to serve a large-scale distributed system. Since a user's d_{sem} lives on exactly one SEM, the following problems are inevitable; temporary denial of service if the network is partitioned, permanent denial of service if SEM suffers from a serious failure and inability to revoke the key pair if an adversary compromises SEM and learn its secrets. To address the problems mentioned, G. Vanrenen et al. proposed a distributed SEM network which acts as SEM. In this paper, we shortly call their system as DSEM. The DSEM consists of trustworthy *islands* distributed throughout Peer-to-Peer(P2P) network. An individual island may still become compromised and reveal its data to the adversary. It may also become unavailable, due to crash or partition. To handle these scenarios, they built *migration* scheme based on threshold cryptography and strong forward security.

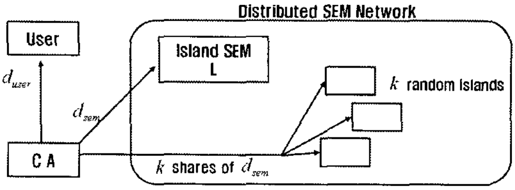

Fig. 1. Key Setup in DSEM.

[Key setup]

Fig. 1 shows key setup procedure in DSEM. Each island acts as a SEM. A CA generates key pair for a user and splits d into two halves. It transmits d_{user} to the user and d_{sem} to an island L. Then, it shares additionally d_{sem} to k islands in the network using threshold cryptography. After those steps are completed, d_{sem} is stored both on the primary island L and on k other islands, so an attacker must either compromise L or compromise t out of the k islands in order to get d_{sem}. Additionally, the shares are proactively updated using proactive secret sharing schemes in[1,2,20,21].

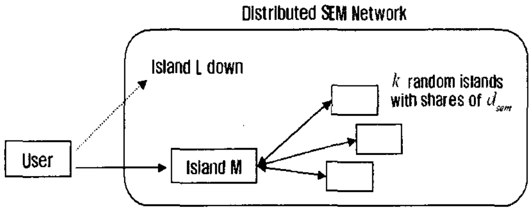

Fig. 2. Migration in DSEM.

[Migration]

If a user issues a request but the island L holding d_{sem} is not available, then the user selects another island M and requests migration. Fig. 2 shows migration procedure in DSEM.

(Step 1) The user connects to another island M instead.

(Step 2) To guarantee strong forward security, the island M generates a new δ in a range $[-r, r]$, and changing d_{sem} to $d_{sem}-\delta$, where r is big enough to keep the key halves changing unpredictably, but small enough to be smaller than d_{sem} and d_{user} for a practically indefinite number of rounds.

(Step 3) M sends δ to the user. Then, the user replaces d_{user} with $d_{user}+\delta$. M splits δ into k shares and sends each to the corresponding d_{sem} shareholder island. Each shareholder island uses its piece to update its share.

(Step 4) Finally, migration is completed and M can then fulfill the user's request.

In migration, M must know $\phi(N)$ to interpolate a polynomial required to perform (k, t)-secret sharing for d_{sem} in (Step 3) and the value r in (Step 2) is ambiguous. The solution for the problems mentioned will be introduced in Section 4.1.

3. NOTABLE PROBLEMS

In this section, we question several inadequate system operations in DSEM and introduce *four requirements* for designing a modified version.

Question 1 : *How can we make k islands perform efficiently a proactive secret sharing ?*

After key setup procedure, k islands periodically participate in a proactive secret sharing to renew periodically their shares for d_{sem} by using the schemes in [1,2,20,21]. However, the schemes in [1,2] cannot be adopted to DSEM, because they are based on the discrete logarithm. Moreover, the scheme in [21] must use $\lambda(N) = lcm(p-1, q-1)$ instead of $\phi(N)$ to make the system proactive. Therefore, it is impossible to make the system proactive without regulating modulus operators. Then, the scheme in [20] can be used in DSEM from the viewpoint of modulus operator. However, DSEM cannot avoid lots of consumption of system resources because it must perform subsharing as many times as the number of shares; each instance of subsharing requires lots of consumption of system resources.

[Requirement 1] *To reduce the overhead caused by subsharing, the system must perform a proactive secret sharing without subsharing.*

Question 2 : *Is DSEM always performed as efficient as SEM ?*

Let us assume the scheme in either[17] or [21] is used for threshold protection. Since the scheme is basically based on (k, k)-additive secret sharing, although the scheme still uses (k, t)-polynomial secret sharing for each share computed by the additive secret sharing to satisfy fault-tolerant property, we can image the following bad situation.

A user's d_{sem} may be shared among k islands by using (k, k)-additive secret sharing. Let a user A's primary island be L_A and a user B's primary island be L_B. Then, k shareholder islands of A's d_{sem} consist of $L_{A1}, L_{A2}, \cdots, L_{Ak}$. We assume that L_B is L_{A4} and both L_A and L_B are eventually compromised at the same time. Then, the following procedure is performed for A to migrate from L_A to M_A successfully:

(Step 1) At least t out of $k-1$ islands, $L_{A1}, L_{A2}, L_{A3}, L_{A5}, \cdots, L_{Ak}$, collaboratively recover the share of L_{A4} by performing an instance of the polynomial interpolation of (k, t)-polynomial secret sharing. To do so, the system must consume lots of communication and computation resources to perform a verifiable recovery protocol[10][18][20][21].

(Step 2) After that, M_A must perform (Step 2) and (Step 3) of migration procedure in Section 2.2.

(Step 3) Of course, B must migrate from L_B to M_B. However, if L_A is L_{B5}, the migration procedure must be more complex.

The main objective of DSEM is to make SEM instantly applicable and scalable. Nevertheless, DSEM cannot present instant cryptographic operation services such as signing or decrypting before finishing the complex procedure mentioned above.

[Requirement 2] *DSEM must be modified to make the cryptographic operation service immediate. That is, the cryptographic operation service, i.e., the signing or decrypting, must be independent of migration.*

Question 3 : *Is a proactive secret sharing meaningful ?*

In DSEM, a user's d_{sem} is stored in the primary island L and shared among k islands. To make shares in k islands robust against adversaries, DSEM performs a proactive secret sharing among k islands. Since a long-term secret d_{sem} is stored in L, the target of adversaries is not one of k islands but L. That is, since the long-term secret d_{sem} is kept in the networking island and the proactive secret sharing does not change it, the execution of the proactive secret sharing cannot contribute to the security of d_{sem}.

[Requirement 3] *Only through all of d_{user}, d_{sem} and k shares for d_{sem} are periodically renewed at the same time, we can make a proactive secret sharing meaningful in DSEM.*

Question 4 : *How many peers are necessary to serve a threshold protection in DSEM ?*

Because DSEM is implemented on JXTA, a peer in P2P acts as an island[16]. Let us consider (k, t)-secret sharing. In usual synchronous communication model, the system allows at most $t-1$ servers to be compromised by an adversary, and needs at least t servers to be correct. That is, k must be greater than or equal to $2t-1$, and at least t server must be available. However, we must consider an inherent property of P2P network such that correct peers in P2P are not always connected to the network. Moreover, an island which acts as a primary island for specific users is also a peer in P2P. So, we must present a solution which prevents a user from performing frequent migration because of simple power down of the primary island without being compromised.

[Requirement 4] *Let Δ be the maximum number of correct peers which are not currently connected to the network. We precisely define the number of servers as $k+\Delta$, where $k=2t-1$. So, we must perform $(k+\Delta, t)$-secret sharing instead of (k, t)-secret sharing to serve a successful threshold protection in stateless model such like P2P.*

4. MODIFICATION OF DSEM

In this section, we present a modified version of DSEM according to four requirements mentioned in Section 3.

4.1 Cryptographic Primitives

First of all, we introduce three cryptographic primitives for our modification as follows.

[N-mRSA]

To minimize the uncertainty of r in migration procedure in Section 2.2 and the insecurity of releasing modulus operator, i.e., $\phi(N)$, for the future use such as proactive secret sharing, we propose a modified version of mRSA. It is based on threshold RSA signature scheme in [6].

[Key setup]

A CA generates a private key d based on the standard RSA. Then, the CA splits the private key into two halves by using $d = d_u^N + d_s^N \bmod N$. It securely transmits d_u^N to the user, and d_s^N to the server.

[Signing]

To sign a message m, the user sends m to the server. Then, the server computes a partial signature $PS_s = m^{d_s^N} \bmod N$ and returns it to the user. Concurrently, the user computes $PS_u = m^{d_u^N} \bmod N$. On receiving PS_s, the user computes a *candidate signature* CS as follows:

$$CS = PS_s \cdot PS_u = m^{d_s^N} \cdot m^{d_u^N} = m^{t \cdot N + d} \bmod N$$

, where $0 \le t < 2$. Finally, the user applies CS to *2-bounded coalition offsetting algorithm* in [6], and computes a valid signature on m.

[Combinatorial Secret Sharing]

In [7,8], L. Zhou et al. proposed *combinatorial secret sharing* which is based on the additive secret sharing. To avoid confusion, they used *share sets* to denote shares of a secret x by using a combinatorial secret sharing and used shares of x only for the values comprising a secret sharing. We can construct share sets, one for each server, of an (k, t)-combinatorial secret sharing from additive secret sharing. To simplify the description, we use abstract modulus operator.

(Step 1) Create $l = \binom{k}{t-1}$ different sets $P_1, \cdots, P_l$ of servers. These sets of servers represent the worst-case failure scenarios: sets of servers that could all fail under the assumption that at most $t-1$ servers are compromised.

(Step 2) Create a sharing $\{s_1, \cdots, s_l\}$ using (l, l)-additive secret sharing scheme. Associate share s_i with failure scenario P_i.

(Step 3) Include secret share s_i in S_p, the share set for a server p, if and only if p is not in corresponding failure scenario P_i. That is, for any server p, share set S_p equals $\{s_i | 1 \le i \le l \wedge p \notin P_i\}$. Note that, by not assigning s_i to any server in a failure scenario P_i, they ensure that servers in P_i do not together have all l shares to reconstruct the secret x. For any set P of servers, the constructed share sets satisfy the following conditions:

- *Condition 1* : $U_{p \in P} S_p = \{s_1, s_2, \cdots, s_l\}$, where $|P| \ge t$.
- *Condition 2* : $U_{p \in P} S_p \subset \{s_1, s_2, \cdots, s_l\}$, where $|P| \le t-1$.

[Server-Assisted Threshold Signature]

In [15], S. Xu et al. proposed a formal method to construct server-assisted threshold signature schemes. It is based on hybrid of threshold signature schemes and two-party signature schemes such as [11,12]. In this paper, we present a practical instance which uses N-mRSA and threshold RSA signature scheme in [6] as cryptographic primitives.

[Assumption]

There are k servers which securely store shares of secret information in the system.

[Key Setup]

A CA generates a private key d based on the standard RSA. Then, the CA splits the private key into two halves by using $d = d_u^N + d_s^N \bmod N$. The CA performs (k, t)-combinatorial secret sharing for d_s^N and generates k share sets. Then, the CA transmits securely d_u^N to the user, and each share set to the corresponding server, respectively.

[Signing]

To sign a message, the user broadcasts a message m to the servers. Then, at least t servers compute collaboratively a partial signature $PS_s = m^{t_1 \cdot N + d_s^N} \bmod N$, where $l = \binom{k}{t-1}$ and $0 \le t_1 < l$. Concurrently, the user computes $PS_u = m^{d_u^N} \bmod N$. On receiving PS_s, the user computes a *candidate signature* CS as follows: $CS = PS_s \cdot PS_u = m^{t_1 \cdot N + d_s^N} \cdot m^{d_u^N} = m^{t \cdot N + d} \bmod N$, where $0 \le t < l+1$. Finally, the user applies CS to $(l+1)$-*bounded coalition offsetting algorithm* in [6], and computes a valid signature on m.

Note that we can compute RSA signatures and perform proactive secret sharings without insecurity of releasing modulus operator, i.e., $\phi(N)$, by using two signature schemes presented above.

4.2 Architecture & System operation

Fig. 3 shows our modified DSEM. In our modified DSEM, there is a peer group(PG) which consists of $(k+\Delta)$ peers, where $k = 2t-1$ and each peer in PG is called as TP-peer. Each TP-peer in PG has *share sets* for all users' d_s^Ns. Then, PG presents cryptographic operation services, i.e., signing or decrypting, to users. Since PG must consist of trustworthy peers, we can depend on *reputation techniques* in P2P for pre-selection of such peers. Each user registers at a single island. The island becomes Home SEM for the user and possesses d_s^N for the user. We present five concrete protocols for our modified DSEM: *key setup, signing, periodical renewal, recovery of compromised islands and recovery of compromised TP-peer.*

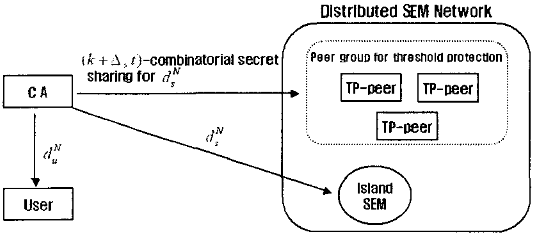
Distributed SEM Network
$(k+\Delta, t)$-combinatorial secret sharing for d_s^N
Peer group for threshold protection
TP-peer
TP-peer
TP-peer
C A
d_s^N
d_u^N
User
Island SEM

Fig. 3. Key setup procedure in our modified SEM.

[Key setup]

A CA generates a private key d based on the standard RSA. Then, the CA splits the private key into two halves as the same as N-mRSA. It transmits d_u^N to the user and d_s^N to Home SEM for the user, respectively. Then, it shares d_s^N among $(k+\Delta)$ TP-peers in PG by using $(k+\Delta, t)$-combinatorial secret sharing. So, our modified DSEM depends on Requirement 4 and the value of Δ can be set according to the system policy.

[Signing]

To sign a message m, a user sends m to Home SEM. Then, both the user and Home SEM perform signing procedure in N-mRSA. Finally, the user can obtain a valid signature on m.

[Periodic renewal]

To renew periodically d_u^N, d_s^N and all share sets for d_s^N, the system performs the following steps.

(Step 1) A user generates a new δ in the range $[-N, N]$. Then, the user performs $(k+\Delta, t)$-combinatorial secret sharing for δ and generates $k+\Delta$ share sets. Then, the user transmits securely each share set to the corresponding TP-peer, respectively. Then, the user computes a renewed half key as $(d_u^N)^{new} = d_u^N - \delta$.

(Step 2) When each TP-peer receives a share set for δ, it adds shares in the received share set to shares in the current share set, respectively. The share sets newly generated can be used to generate a renewed half key as $(d_s^N)^{new} = d_s^N + \delta$. Then, at least t out of $k+\Delta$ TP-peers send securely their renewed share sets for $(d_s^N)^{new}$ to Home SEM for the user.

(Step 3) On receiving at least t share sets, Home SEM can combine them and compute the renewed half key $(d_s^N)^{new}$. After that, Home SEM generates a random string rs and computes $challenge = rs^{(d_s^N)^{new}} \bmod N$. Then, Home SEM sends rs and *challenge* to the user.

(Step 4) The user computes $response = rs^{(d_u^N)^{new}} \bmod N$ and combines it with *challenge* for generating a RSA signature for rs. After that, the user checks the validity of the RSA signature. If the result is successful, the user sends *success notification* and *response* to Home SEM and replaces d_u^N with $(d_u^N)^{new}$. Otherwise, the user sends *error notification* to Home SEM.

(Step 5) If Home SEM receives *success notification* from the user, it also checks the validity of *response* through the same way that the user performed. If the result is successful, Home SEM replaces d_s^N with $(d_s^N)^{new}$ and finishes the procedure. Otherwise, it accuses TP-peer, who sent an erroneous share set, to PG by broadcasting an *accusation message*. Then, Home SEM tries to perform (Step 3) again by using another shares in the received share set.

By using a simple challenge/response, the periodic renewal can be verifiably performed. During the run of periodic renewal, the user can perform signing with old keys, d_u^N and d_s^N, in contrast with DSEM. Our modified DSEM guarantees therefore Requirement 2. As you have seen, the periodic renewal depends on both Requirement 1 and Requirement 3.

[Recovery of compromised SEM]

For successful recovery of a compromised SEM from adversaries, the system performs the following steps.

(Step 1) To sign a message m, a user sends it to Home SEM and requests a partial signature PS_s.

(Step 2) During the specific time bound, if the user does not receive PS_s from Home SEM, the user broadcasts m and an accusation message to PG. Then, the user can compute a valid signature on m by using the *server-assisted threshold signature* between the user and PG.

(Step 3) If the number of accusation messages from users exceeds a specific limit based on the system policy, Home SEM is rebooted and initialized by clean copy.

(Step 4) At least t TP-peers send securely their share sets to Home SEM. Then, Home SEM can obtain l shares to reconstruct d_s^N.

(Step 5) After that, the user performs *periodic renewal* mentioned before.

During the recovery of the compromised SEM, the user also performs cryptographic operation such as signing or decrypting via server-assisted threshold signature scheme in Section 4.1, although the performance is lower than N-mRSA. Therefore, our modified DSEM also guarantees Requirement 2 in spite that Home SEM is compromised.

[Recovery of compromised TP-peer]

When every TP-peer in PG receives accusation messages exceeding a specific limit from "island SEMs", which occurs in (Step 5) of periodic renewal, the system performs the recovery procedure for the compromised TP-peer.

(Step 1) The accused TP-peer is rebooted and initialized by clean copy.

(Step 2) Each TP-peer except the accused TP-peer sends shares, which are owned by both each TP-peer and the accused TP-peer, to the accused TP-peer, respectively.

(Step 3) Then, the accused TP-peer can reconstruct share set for itself. If the accused TP-peer wants to verify the validity of the reconstructed share set, given for a challenge message m, both Home SEM and the accused TP-peer check the equivalence of a generated partial signature on m by collaborating with another $t-1$ TP-peers.

(Step 4) The user is recommended to perform *periodic renewal*. However, the user will perform periodic renewal in the near future without the recommendation.

During the proactive activities such as periodic renewal and two recovery protocols in our modified DSEM, the system does not perform subsharing for each share. That is, our modified DSEM guarantees Requirement 1.

5. DISCUSSION

In this section, we discuss the security of our proactive scheme(i.e., periodic renewal and two recovery protocols) in Section 4.2 and the notable features of our modified DSEM.

5.1 Security of Proactive Scheme

Now, we discuss the security of our proactive scheme: periodic renewal and two recovery protocols. For the security of both N-mRSA and an instance of server-assisted threshold signatures in Section 4.1, please refer to [6,15]. A user's half secret, d_s^N, is secretly shared among TP-peers in PG, and renewed or recovered by the schemes in Section 4.2. Then, they must satisfy the following properties to be secure proactive secret sharing scheme.

◆ *Independency* : New shares for the secret cannot be combined with old shares to reconstruct the secret.

◆ *Secrecy* : The secret remains unknown to adversaries.

◆ *Availability* : Correct servers together have sufficient shares of the secret to reconstruct it.

We assume *short-term constrained adversary* introduced in [6] to characterize adversary; given that time is divided into periods, the adversary cannot break t or more servers during any time period, where the number of server is $k+\Delta$, $k=2t-1$ and Δ is the maximum number of correct peers which are not currently connected to the network. Now, we show simply that our proactive scheme satisfies independency, secrecy and availability properties.

◆ *Independency*

: After a user generates a random value, δ, in (Step 1) in Periodic renewal, the random value can be used to renew shares for both d_u^N and d_s^N. In contrast that the existing proactive secret sharings used in the original DSEM do not change d_s^N but shares for d_s^N, our periodic renewal changes/renews the secret itself, i.e., d_s^N. So, the shares in PG during a time period can be only used to reconstruct d_s^N in the time period. Therefore, an adversary who even knows d_s^N in a time period without keeping corruption of at least t servers (i.e., who succeeds in corrupting Home SEM of the user) cannot know the newly renewed d_s^N in the next time period. Our proactive scheme satisfies therefore *Independency.*

◆ *Secrecy*

: To show Secrecy, it suffices to show that an adversary cannot obtain all l shares by corrupting at most $t-1$ TP-peers in a time period. Let P be the set of TP-peers corrupted in a time period, $|P|\le t-1$ holds. Due to *Condition 2* in the Section 4.1, there is at least one share which the adversary cannot obtain. Our proactive scheme satisfies therefore *Secrecy.*

◆ *Availability*

: To show Availability, it suffices to show that correct servers can reconstruct d_s^N. In a time period, there are at least t correct TP-peers in PG connected in the network because of Requirement 4. Let P be the set of correct TP-peers connected in the network, $|P|\ge t$ holds. Due to *Condition 1* in the Section 4.1, correct TP-peers can collect l shares for reconstructing d_s^N. So, the correct TP-peers can perform recovery of compromised SEM or TP-peer. That is, our proactive scheme satisfies *Availability.*

In [13], S. Jarecki et al' introduced the weakness of proactive scheme of threshold RSA in [6]. It is the basis of cryptographic primitives in Section 4.1. However, since our proactive scheme does not depend on subsharing in contrast to the scheme in [6], an adversary in [13] cannot succeed in learning the private exponent d; i.e., the adversary can learn at most $\lg(k+\Delta)$ most significant bits(MSBs) of d during the entire life-time. So, we do not need to consider the weakness introduced in [13].

5.2 Notable Features

Our modified DSEM has the same features as the original SEM, because it succeeds to the characteristics of the original SEM. Moreover, our modified DSEM solves the problems mentioned in Section 3 with the following desirable features.

◆ *Removal of both insecurity of releasing $\Phi(N)$ and uncertainty of r*

: Our modified DSEM uses three cryptographic primitives in Section 4.1 which are based on N for modular operator of RSA exponent.

◆ *Efficient and timely signing or decrypting*

: In the DSEM, the user cannot perform signing or decrypting until the migration is finished. On the other hand, the user can still perform signing or decrypting via server-assisited threshold signature, in spite that either periodic renewal or recovery is under way in our modified DSEM. That means the capability for signing or decrypting is independent of periodic renewal or recovery.

◆ *Strong against denial of service attack*

: Our modified DSEM is strong against denial of service attack by using an alternative operation, i.e., server-assisted threshold signature, although the performance is lower than N-mRSA. That is, the user can still perform siging or decrypting in spite that the user's Home SEM is compromised.

◆ *Meaningful proactive secret sharing*

: In contrast to DSEM, our modified DSEM can appropriately renew a user's half, d_u^N, the corresponding half of SEM, d_s^N, and shares for the half of SEM per time period for renewal.

◆ *Simplified renewal and recovery*

: In DSEM, they used well-known proactive secret sharing schemes such as [20,21] to perform periodic renewal, recovery of a compromised island or migration. The schemes referred must require lots of system resources to perform subsharing and verifiable secret sharing. However, since our modified DSEM does not require any subsharing and verifiable secret sharing, it consumes the minimized system resources. Such the simplified renewal and recovery can be achieved by adopting combinatorial secret sharing, user intervention and simple challenge/response.

In [14], S. Koga et al' proposed a solution to prevent denial of service attack by picking out malicious users' requests though one-time ID. Since their solution did not consider the possibility of the corruption of SEM, it did not present a solution for recovering the compromised SEM. Nevertheless, we believe that S. Koga et al.'s proposal can be used for supporting authentication of users' requests in our modified DSEM.

6. CONCLUSION

In this paper, we reviewed G. Vanrenen et al.'s distributed SEM approach and presented a practical model for actual implementation. Our modified DSEM succeeds to the advantages of the original SEM and also provides desirable features comparing with G.Vanrenen et al.'s proposal.

7. REFERENCES

[1] A. Herzberg, M. Jakobsson, S. Jarechi, H. Krawczyk, and M. Yung, "Proactive public key and signature systems," *ACM Conference on Computer and Communications Security*, pp.100-110, 1997.

[2] A. Herzberg, S. Jarecki, H. Krawczyk, and M. Yung, "Proactive secret sharing or: How to cope with perpetual leakage," *Advanced in Cryptology-CRYPTO 95*, LNCS 963, pp.339-352, 1995.

[3] C. Adams and S. Lloyd, *Understanding public-key infrastructure: concepts, standard, and deployment considerations*, Indianapolis: Macmillan Technical Publishing, 1999.

[4] D. Boneh, X. Ding, G. Tsudik, and C. M. Wong, "A method for fast revocation of public key certificates and security capabilities," *10th USENIX Security Symposium*. pp. 297-308, 2001.

[5] G. Vanrenen and S.W. Smith, "Distributing Security-Mediated PKI," *1st European PKI Workshop Research and Applications*, LNCS 3093, pp.213-231, 2004.

[6] Haiyun Luo and Songwu Lu, "Ubiquitous and Robust Authentication Services for Ad Hoc Wireless Networks," *UCLA Computer Science Technical Report 200030*, Oct. 2000.

[7] Lidong Zhou, Fred B. Schneider, and Robbert van Renesse, "COCA: A Secure Distributed Online Certification Authority," *ACM Transactions on Computer Systems*, pp.329-368, 2002.

[8] Lidong Zhou, "Towards Fault-Tolerant and Secure On-line services," *PhD Dissertation, Department of Computer Science, Cornell University*, Ithaca, NY USA. April, 2001.

[9] M. Naor and K. Nissim, "Certificate revocation and certificate update," Proceedings *7th USENIX Security Symposium, San Antonio*, Texas, pp.217-228, 1998.

[10] P. Feldman, "A Pracitcal Scheme for Non-Interactive Verifiable Secret Sharing," *Proc. of 28th FOCS*, 1987.

[11] P. MacKenzie and M. Reiter, "Networked Cryptographic Devices Resilient to Capture," *IEEE Security and Privacy'01*, 2001.

[12] P. MacKenzie and M. Reiter. "Two-Party Generation of DSA Signatures," *Crypto'01*, LNCS 2139, pp.137-154, 2001.

[13] S. Jarecki, N. Saxena, and J. H. Yi, "An Attack on the Proactive RSA Signature Scheme in the URSA Ad-Hoc Network Access Control Protocol," *ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN)*, pp.1-9, 2004.

[14] S. Koga, K. Imamoto, and K. Sakurai, "Enhancing Security of Security-Mediated PKI by One-time ID," *4th Annual PKI R&D Workshop*, NIST, USA, 2005.

[15] S. Xu and R. Sandhu, "Two Efficient and

Provably Secure Schemes for Server-Assisted Threshold Signatures," *CT-RSA*, 2003.

[16] Sun Microsystems, Inc, *Project JXTA : Java Programmers Guide*, 2001.

[17] Tal Rabin, "A Simplified Approach to Threshold and Proactive RSA," *Advanced in Cryptology-CRYPTO 98*, LNCS 1462, pp.89-104, 1998.

[18] Torben Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," *Advanced in Cryptology-CRYPTO 91*, pp.129-140, 1991.

[19] X. Ding, D. Mazzocchi, and G. Tsudik, "Experimenting with server-aided signatures," *Network and Distributed Systems Security Symposium*. 2002.

[20] Y. Frankel, P. Gemmell, P. MacKenzie, and M. Yung, "Optimal resilience proactive public key cryptosystems," *IEEE Symposium on Foundations of Computer Science*, pp. 440-454, 1997.

[21] Y. Frankel, P. Gemmell, P. MacKenzie, and M. Yung, "Proactive RSA," *Advances in Cryptology-CRYPTO 97*, LNCS 1297, pp.440-454, 1997.

Jong-Phil Yang

He has completed his B.S and M.S. degrees in the Department of Computer Science from PuKyong National University in 1999 and 2001, respectively, and Ph. D. degree in the same department and university in 2005. He is currently a visiting scholar in the Department of Computer Science and Communication Engineering of Kyushu University, Japan. His interests are ubiquitous computing security, PKI, secret sharing and anonymity, etc.

Mi-Sun Shim

She received the B.S. degree in the Department of Computer Science from PuKyong National University in 2003. She has also completed her M.S. degree in the Department of Information Security from PuKyong National University in 2005. She is currently working as a SW engineer in INFOSEC Technologies. Her interests are cryptograpic protocols, secret sharing and system security, etc.

Kyung Hyune Rhee

He has completed his Masters of Science (M. Sc) and Doctorate degree (Ph. D) in the Department of Applied Mathematics from Korea Advanced Institute of Science and Technology (KAIST), Republic of Korea 1985, 1992 respectively. He served as a visiting professor in Department of Information and Computer Science, University of California at Irvine (UCI), USA during 2001 through 2002. He also served as a faculty consultant in Colombo Plan Staff College, Manila, Philippines during 2002 through 2003. He also served as an academic director in Korea Multimedia Society (KMMS) 어ring 1987 through 2002, and has been working as a finance director since 2002. He is currently a full professor at the Division of Electronic, Computer and Telecommunication Engineering, PuKyong National University (PKNU), Busan, Republic of Korea.

His main interests are mobile and ad-hoc security, ubiquitous computing, cryptographic protocols and statistical analysis of cryptographic algorithms, etc.