Alexandria Engineering Journal (2012) 51, 185–191

Alexandria University

Alexandria Engineering Journal

www.elsevier.com/locate/aej
www.sciencedirect.com

ORIGINAL ARTICLE

A model to authenticate requests for online banking transactions

Saad M. Darwish *, Ahmed M. Hassan

Department of Information Technology, Institute of Graduate Studies and Research, Alexandria University, 163 Horreya Avenue, El Shatby 21526, P.O. Box 832, Alexandria, Egypt

Received 16 March 2011; revised 9 December 2011; accepted 28 February 2012
Available online 12 April 2012

KEYWORDS

Identity Based Encryption (IBE);
Electronic transactions;
Online banking security

Abstract As the number of clients using online banking increases, online banking systems are becoming more desirable targets for attacks. To maintain the clients trust and confidence in the security of their online banking services; financial institutions must identify how attackers compromise accounts and develop methods to protect them. Towards this purpose, this paper presents a modified model to authenticate clients for online banking transactions through utilizing Identity-Based mediated RSA(IB-mRSA) technique in conjunction with the one-time ID concept for the purpose of increasing security, avoiding swallow's sorties and preventing reply attacks. The introduced system exploits a method for splitting private keys between the client and the Certification Authority (CA) server. Neither the client nor the CA can cheat one another since one-time ID can be used only once and each signature must involve both parties. The resulting model seems to be practical from both computational as well as storage point of view. The experimental results show the effectiveness of the proposed model.

1. Introduction

Electronic banking that allows people to interact with their banking accounts via the Internet from virtually anywhere in the world provides enormous benefits to consumers in terms of the ease and cost of transactions. This system permits consumers to request information and carry out most of banking services such as balance reports, inter-account transfers, and bill payment. The basic architecture of online banking system consists of three major components [1]: (1) Client; (2) Application server that takes care of the server script and checks for the ODBC connectivity for mapping to the database in order to fulfill client and administrator's requests; and (3) Database, which stores client and bank data.

* Corresponding author. Tel.: +20 1222632369; fax: +20 3 4285792.
E-mail addresses: saad.darwish@gmail.com, saad.darwish@alex-igsr.edu.eg (S.M. Darwish), Ahassan1968@hotmail.com (A.M. Hassan).

Peer review under responsibility of Faculty of Engineering, Alexandrian University.

ELSEVIER Production and hosting by Elsevier

1110-0168 © 2012 Faculty of Engineering, Alexandria University. Production and hosting by Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.aej.2012.02.005

While online banking offers enormous advantages and opportunities, it faces different kinds of risks that are specific to conduct sensitive business over the Internet. So, it is imperative that banks implement strong security approaches that can adequately address, monitor, manage and control risks and security threats to the bank. Security aims to prevent fraudsters from accessing online banking accounts that don't belong to them, and subsequently viewing confidential information, causing malicious damage and stealing funds. In order to provide effective and secure banking transactions, there are two technology issues needed to be resolved [2]: (1) Security: is the primary concern of the Internet-based industries. The lack of security may result in serious damages and (2) Authentication: Encryption may help make the transactions more secure, but there is also a need to guarantee that no one alters the data at either end of the transaction.

In general, the solutions to the online banking security issues require the use of software-based systems or hardware-based systems or a hybrid of them. The software-based solutions involve the use of encryption algorithms while hardware-based solutions involve the use of devices such as the smartcard. Software-based protection is easily obtained at lower costs than hardware-based protection. Consequently, due to the easy portability and ease of distribution through networks; software-based systems are more abundant in the market. However, software-based protection has many potential hazards like attacking the encryption algorithms by means of brute force and analysis attacks [1–3].

Encryption, which modifies information in a way that makes it unreadable until the exact same process is reversed, is one of the methods used to solve the problem of attack trees, which identify how attackers compromise accounts and develop methods to protect them [4]. In the literature, there exist many algorithms available to implement the public key cryptography, like Rivest–Shamir–Adleman (RSA) encryption, Advanced Encryption Standard (AES) and Data Encryption Standard (DES). However, these techniques are slow when large volumes of data are to be encrypted [5]. Other encryption techniques include mediated RSA (mRSA) [6], which is a simple and practical method of splitting RSA private keys between user and the online trusted server. Both the user and the trusted server employ their respective half-keys in a way that is functionally equivalent to (and indistinguishable form) standard RSA. Neither the user nor the server knows the factorization of the RSA modulus and neither can decrypt/sign message without the other's help. The main problem of mRSA encryption approach is that it still relies on public key certificates to derive private/public keys. This leads to the issues of certificate management like revocation, distribution, storage and verification. Refer to [7,8] for a detailed description and security analysis of the mRSA algorithm.

Recently, Identifier-Based Encryption (IBE) is emerged as a cryptographic scheme in modern secure banking systems to protect online transactions. Identity-based public key encryption facilitates easy introduction of public key cryptography by allowing an entity's public key to be derived from arbitrary identification values, such as name or e-mail address. The main practical benefit of identity-based cryptography is in greatly reducing the need for, and reliance on, public key certificates and solving certain public key management problems since there is no need to maintain a great database containing a list of public keys and their respective owner [7]. A more recent work is proposed to combine identity-based encryption with mRSA in one framework, IB-mRSA, to improve security against adaptive chosen cipher attacks that represent the immense threat in RSA algorithm. IB-mRSA protocol allows the sender (encryptor) to skip the costly checking of individual public key certificates. Furthermore, for real time applications IB-mRSA algorithm takes roughly 4–5 times less than plain RSA technique [8].

The major weakness of the IB-mRSA based online banking systems is that their disability to prevent reply attacks and their vulnerability against denial of the services (DoSs) attacks since an attacker can send many requests to the trusted server. In the literature, there are three types of DoS attacks [9]: against server's bandwidth, memory, and CPU. The purpose of the first attack is that a server cannot receive any more messages. The second one is performed to make a server stores large quantities of waste states. The last one is the attack, which makes a server computes a lot of quite inefficient processing. Most previous security researches focus on DoS attacks against server's CPU. To handle the problem of DoS attack, the idea of employing one-time ID is suggested in order to make the attackers cannot reuse the requests generated by legal user because one-time ID dynamically changes.

1.1. Contribution

Our contribution in this paper is to blend the attractive features of IB-mRSA protocol with one-time ID based DoS prevention technique in an integrated model to authenticate requests for online banking transactions. The proposed model represents a variant of IB-mRSA technique that can avoid DoS attacks and prevent leakage of user's identity. The rationale of using IB-mRSA protocol is that it combines the advantages of fast revocation and identity based public keys, which makes it difficult to control a user's security privileges for identity revocation.

This model is simple, secure and very efficient to protect the privacy of clients since one-time ID can be used only once. Moreover, the established model gives extra security against hackers to guess private keys and possesses the ability of supporting revocation not just for signatures but also for (public key) encryption as well. Both the architecture and an implementation of this model are discussed as well as the performance, compatibility, and usability aspects.

The rest of the paper is organized as follows. Next section provides a brief synopsis of our work, its contribution and describes the proposed security model in details. Subsequently, implementation and performance measurements are discussed in Section 3; it is followed by efficiency and security analysis in Section 4. In Section 5, the paper concludes with the summary of benefits of the presented model.

2. Methodology

This section describes the proposed model that is derived from Rajalakshmi et al. approach [8] with some necessary modifications to satisfy our new requirement such as certainty and simplicity in the dispute resolution and avoiding swallow's attacks (generating a signature for any other message using the proxy's one-time public key). The major difference between the offered model and the original technique is through utilizing a small computational complexity method used for one-time ID calculation employed in order to reduce damage of possible attacks. Fig. 1 shows the diagram of the proposed model that depends on three modules.

2.1. Certification Authority (CA) module

Certification Authority (CA) module is the central of the whole architecture corresponding to a trusted third party that issues certificates. The purpose of CA server is to generate client bundle and Security Mediator (SEM) bundle that encapsulate the needed security component for electronic banking transactions by splitting private keys into two parts, one for the client and the other for the SEM. It also issues the system-wide certificate containing the common modules *n*. The CA server is isolated from the Internet to prevent unauthorized accesses using firewall.

Inside the suggested model, to sign or decrypt a message the client must obtain a specific token message from the CA server. Without this token the client can not use his private key. Furthermore, to revoke the client's ability to sign a message, the security administrator instructs the CA server to stop issuing tokens for client's public key. The CA architecture is transparent to the sender of a message and to the verifier of a signature because the encryption and verification operations remain the same as in classical RSA. In this state, the use of CA's architecture removes the need to enquire about the status of a public key before using it.

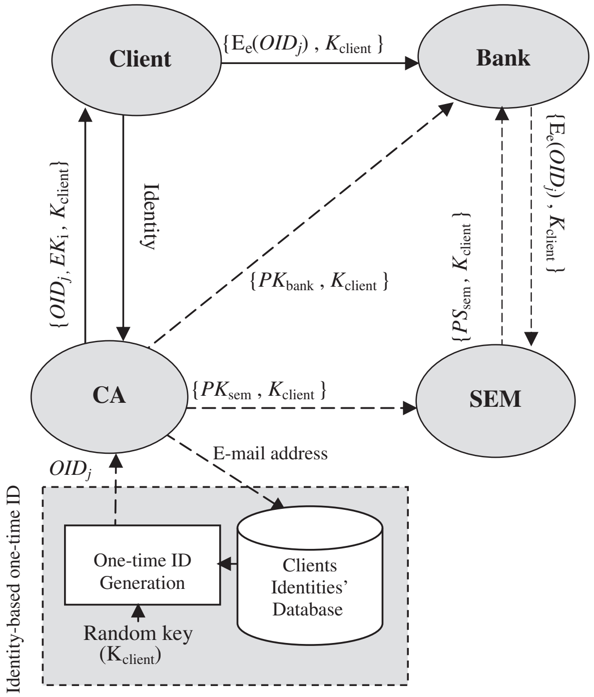
Client
Bank
$\{E_e(OID_j)$, $K_{client}\}$
Identity
$\{OID_j, EK_i, K_{client}\}$
$\{PK_{bank}$, $K_{client}\}$
$\{PS_{sem}, K_{client}\}$
$\{E_e(OID_j)$, $K_{client}\}$
CA
SEM
$\{PK_{sem}$, $K_{client}\}$
E-mail address
OID_j
One-time ID Generation
Clients Identities' Database
Random key (K_{client})
Identity-based one-time ID

Figure 1 The proposed online transactions security model. OID_j is the Client's one time ID with *j*th session; *EK* is the Public key of the client; PK_{bank} is the Partial private key of a bank; PK_{sem} is the Partial private key of a SEM; PS_{sem} is the Partial signature of a SEM; PS_{bank} is the Partial signature of a bank and K_{client} is the Random number for each session.

2.2. Security Mediator (SEM) module

Security Mediator (SEM) module is an online partially trusted server. The SEM can eliminate the need for certificate revocation list since the private key operations can not occur after revocation. A SEM can be configured to operate in a state or stateless model. The former involves storing per user state (half-key and certificate) while, in the latter, no per user state is kept; however, some extra processing is incurred for each user request. The trade-off is clear: the former and fast request handling versus the latter and somewhat slower request handling. See [9] for more details.

2.3. One-time ID module

One-time ID Module is a user's extraordinary identity, which has two properties: (i) an attacker cannot specify who is communicating even when he eavesdrops on one-time ID, and (ii) one-time ID can be used only once [10]. To realize perfect forwarded security, our model employs an efficient public mapping function that transforms from identity string to one-time ID. In general, using one-time ID allows us to eliminate one more point of vulnerability.

In this work, we assume that: (1) the communication channel between each client and a CA server is reliable (but neither private nor authenticate). Reliability of the channel implies that the underlying communication system provides sufficient error handling to detect all corrupted packets, and handles both of timeouts and retransmission. Furthermore, even if the client is disconnected from the network after sending a signature request to its CA server and before receiving a reply, the client will finally obtain the correct reply whenever the channel is re-established; (2) communication between the client and the bank does not have to be protected. An attacker can eavesdrop and modify the contents of the request message; (3) as malicious users may employ CA server as a signature oracle to answer their queries. Hence, the proposed protocol requires that CA's signature scheme be secure against adaptive chosen message attacks in which the attacker can ask a signature oracle to sign arbitrary message based on the outcome of previous signatures queries [3]. So that the queries will not be injured to forge CA's signatures.

We now turn to the detailed description for the proposed model that involves three algorithms: key generation, signing and verifying (see Fig. 1).

2.3.1. Key generation

Step 1 The client first sends a message to the CA containing the client's identity (*ID*). We use e-mail addresses as a unique identifier of clients to the system that will be used to compute the public exponent for each one.

Step 2 The CA server checks the client‘s identity. Only if this identity is valid according to the sorted data, the CA takes care of all key setup. It generates a distinct set $\{p_i, q_i, e_i, d_i, d_i^{sem}\}$ for each client *i*. The first four parameters are generated in the same manner as in standard IB-mRSA [6,8], in which a public key EK_i is computed as:

$$EK_i = (n_i, e_i) \tag{1}$$

the modulus n_i is a product of two larger randomly chosen primes p_i and q_i, and the public exponent e_i is constructed as the output of $F(ID_i)$, where F is a mapping function. Here, F is represented as a binary string of the same length as the RSA modulus with the least significant bit set:

$$e_i \leftarrow F(ID_i) = 0\|F(ID_i)\|1 \tag{2}$$

In addition, CA produces a corresponding private key

$$\begin{aligned} &PK_i = (n_i, d_i), \\ &d_i * e_i = 1(mod(\phi(n_i)), \\ &\phi(n_i) = (p_i - 1)(q_i - 1) \end{aligned} \tag{3}$$

in this situation, no one has possession of d_i. Instead d_i is effectively split into two parts d_i^{bank} and d_i^{sem} in which the bank and a SEM respectively secretly hold. The relationship among them is:

$$\begin{aligned} &d_i = d_i^{sem} + d_i^{bank} \bmod \phi(n_i), \\ &PK_{bank,i} \leftarrow (d_i^{bank}, n_i), \\ &PK_{sem,i} \leftarrow (d_i^{sem}, n_i) \end{aligned} \tag{4}$$

note that, knowledge of a half key cannot be used to derive the entire private key. Therefore, neither the bank nor the SEM can decrypt or sign a massage without mutual consent.

2.3.2. Signing

Step 3 At the first session, the CA generates the list of clients' identity and stores it in encrypted database for security. This identity is later used to create each client one-time ID for jth session (OID_j). The OID_j is derived as follow:

$$OID_{i,j} = H(ID_i, K_{client}) \tag{5}$$

where H is a collision-resistant hash function and K_{client} is a random key generated for each client in each session. This shared secret key is required between the CA and SEM for state maintenance. The hash function is a one-way function that maps a variable-length message into a fixed-length value called a hash code. Because hash functions are typically quite complex, it is useful to examine some very simple hash functions to get a feel for the issues involved like fast computation. In this paper, a low-cost Secure Hash Algorithm (SHA-1) [11] is used that provides data integrity. At this point, the OID updating is taking place via using K_{client}.

Step 4 Following the computation of the above values, CA issues to the client the public key certificate, EK_i, plus the token ($OID_{i,j}$, K_{client}) created for each session. In addition, the package ($PK_{sem,i}$, K_{client}) is securely communicated to the SEM and the partial private key's bundle ($PK_{bank,i}$, K_{client}) is safety forwarded to the bank.

Step 5 After receiving the token from CA, the client sends a request to the bank by launching a message m that includes the client one-time ID encrypted with the received public key along with K_{client} that is used to verify the client.

$$m \leftarrow E_{e_i}(OID_{i,j}),\ k_{client} \tag{6}$$

The bank receives the above response, confirms that the content of a response is fresh by checking K_{client}, and then forwards the received message to the SEM for authentication (i.e. legal request).

2.3.3. Verifying

Step 6 SEM checks that the client is not revoked by comparing both of k_{client}^{bank} and k_{client}^{CA} getting from bank and CA respectively. If so, it signs the requested message with its private key to produce a partial signature $PS_{sem,i} = m^{d_i^{sem}} modn_i$ and replies with it to the bank. If the client is revoked, the SEM gives an error.

Step 7 The bank receives $PS_{sem,i}$ that is the token enabling signature generation and computes:

$$\begin{aligned} &PS_{bank,i} = m^{d_i^{bank}} modn_i, \\ &m' = (PS_{sem,i} * PS_{bank,i})(modn_i) \end{aligned} \tag{7}$$

here, m' is the plain signature. If the bank's half signature fails verification (i.e. it signs a different message or includes an incorrect signature counter), the bank abort the protocol and concludes that a hostile attack has occurred. If there is no error, the banking system allows the client to perform its online transactions. In this case, no signature storing is required for the signer to prove the server's cheating.

As follows from the protocol description above, both CA and SEM maintain state. The state for client i kept by the SEM is K_{client} that is critical in authenticating client's requests. Hence the integrity of these values should by protected by SEM against illegal tampering. On the other hand, the CA state amount to the user's identity that should be kept secret.

3. Experimental results

To evaluate the proposed model and to obtain valuable experimental and practical data, we implemented the entire model as a fully functional package. Programming on this model is component-based. The only thing we need to implement is to write our components and wire them with the existing components that can perform networking operation using specific TCP/IP protocol.

In the software of the suggested model, there exist three components we will use (as depicted in Fig. 1): (1) CA and administration utility that includes certificate issuance and revocation interface, (2) SEM structure that takes account of verification and partial signature operations; and (3) client libraries that contains encryption function. We ran each component on a server equipped with 1.2 GHz Intel Pentium 4 processor and 1 GB of RAM. In this platform, the signed e-mail can be verified by any S/MIME capable e-mail client such as Microsoft Outlook. All test machines ran Windows NT version with all non-supplementary services turned off. All experiments were conducted over a 100 Mbps Ethernet LAN in a lab setting. We measured the roundtrip latency between the two machines at 5 milliseconds (ms), and the maximum sustained throughput of the network link at 7 Mbps.

Before implementing the proposed model, we should decide on the value of various model's parameters such as the length of signature, the length of random numbers and hash values, and finally the value of both n and e. These parameters should be selected consistent with the desired security level for an application. As stated in [12], for 80-bit security that might be considered as the minimum today for strong security, the signature length should be at least 1024 bits. Assuming that the hash function is a secure one, in our model 80-bit length random number and hash outputs are good enough for 80-bit security. The value of n and e were chosen to be 40 bits since choosing fewer values might cause some security vulnerability.

To validate the goals and experiment with the proposed model's implementation, we ran a number of tests with different key sizes. In the first experiment, we measured communication latency by varying the K_{client} size, which directly influences message sizes. Herein, the size of the requested signature is determined by the digest size of the hash function. The results are shown in Table 1. Latency is calculated as the round-trip delay between the client and the CA. Each experiment involved 20 iterations and all reported timings are in milliseconds. As Table 1 illustrates, despite large variances in the key size, the difference in the model's latency time is small. With an 8192-bit key, the model is fast enough to send its signature in roughly 36 ms.

In the second experiment, the IB-mRSA results are obtained by measuring the time staring with the sending of client's identity message to the CA and ending with the client authentication by the bank. The measurements are illustrated in Table 2, which are taken with the SEM operating in a stateful mode. For all key sizes, the timing is determined by the sum of the round-trip between client CA packet delay and the service time at both CA and SEM. With a 1024-bit key, the system prototype is fast to finish its verification in 81.3 ms. We believe that the model's structural design is suitable for online banking system.

The third experiment investigates the problem of client's authentication and verification, where the on hand system prototype must decide based on $PS_{bank,i}$ and m' if the client with the claimed signature is in fact from this client or whether it is a fraud client. The evaluation is based on varying the client key size, K_{client}, and the corresponding FAR (how frequently imposter clients are accepted by the system) and FRR (determines how often clients are rejected from the system.) are then calculated. The test set consists of two sets. The first set is the set of clients, where the system was trained on their correct identity. The second set is the set of imposters that is formed of additional 200 signatures unknown to the system. Results shown in Table 3 indicate that the proposed model performs very well on both tasks of accepting clients and rejecting imposters. The IB-mRSA' authentication used in our program has some variations in key lengths.

In the last experiment, in order to compare the performance of our model and the SAS model elaborated by Ding et al. [9], we implemented both schemes and measured the computation's time delays. Table 4 summarizes our findings in the experiment. These experimental results show that the proposed model offers a substantial computational advantage over SAS with respect to client's computation. Moreover, the total time required to verify a signed message in the proposed model is at least twice the time in SAS model in spite of greater bandwidth utilization. This is because SAS is a three-round while our model is five-round authentication protocol.

Table 1 Communication latency.

K_{client} size (bit)	Message size (byte)	Average latency (ms)
512	102	5.1
1024	167	6.3
2048	296	10.8
4096	588	28.2
8192	1365	36.6

Table 2 IB-mRSA measurements.

K_{client} size (bit)	Authentication time (ms)
512	15.2
1024	81.3
2048	535.4
4096	3690.5
8192	15965.6

Table 3 Summary of experimental results.

K_{client} size (bit)	FRR (%)	FAR (%)
512	0.003	≅0
1024	0.003	≅0
2048	0.002	≅0
4096	0.001	≅0
8192	≅0	≅0

4. Analysis

We now consider the efficiency and security aspects of the proposed electronic banking transactions model.

4.1. *Efficiency*

As verified from the experimental results, the suggested model significantly speeds up client's authentication computation for slow, resource-limited devices and preserves CPU resource. In general, the cost of our signature model can be broken up as follows: (1) Network overhead: round-trip delay between client and CA; (2) CA computation: signature computation plus other overhead (including one-time ID calculation and database processing). Heuristically CA's computational dominates the overall signature generation cost. Note that CA is high-ended server, if needed it might even be equipped with specialized cryptographic hardware/firmware; and (3) SEM computation: verification of the SEM half-signature and other overhead (e.g. legal request checking). Clearly, as compared with the signature method introduced in [8], the extra cost is only in Step 2, which consists of hash operation. In general, hash computations are negligible as compared to public key operations.

Another important issue regarding the model's efficiency is the CA throughput. Undoubtedly, a real-world CA must be highly fast, optimized fault tolerance and of course highly secure. Our initial prototype offers most of these properties and much work remains to be done in this regard. Compared

Table 4 Experimental comparison of two models (ms).

	SAS model	Proposed model
Client's computation	7.34	3.3
Signing computation	6.9	37.5
Verifying computation	7.8	15.4

with previous signature approach [8,9,12] regular clients pay much fewer CPU cycles while they are well protected with more security strength. The proposed model affords some interesting security features as discussed in Section 4.3.

There is a trade-off between our model and the SAS model (see Table 5). On one hand, our work reduces the signature size largely and avoids the complexity and cost of one-time based key generation algorithm. On the other, it requires a longer signature and more expensive authentication algorithm.

Table 6 once more makes a comparison between the two protocols but now in terms of communication's efficiency. As seen from this table, SAS provides more efficiency with respect to number of rounds required for authentication with a decrease in the length of the exchanged messages. However, in SAS, it becomes possible to produce fraudulent user signatures since state is kept of all prior SAS signatures, so the adversary can sign on behalf of server. The proposed model eliminates such point of vulnerability, prevents reply attacks via employing one-time identification, and thwarts adaptive chosen ciphertext attacks in which the cryptanalyst chooses adaptively ciphertext depending on the result of prior decryptions and causes it to be decrypted with an unknown key.

The comparison of the two models in terms of network delay is formally discussed as follows. Let the network delay is composed of several delay elements expressed as [12]:

$$Delay_{Network} = D_t + D_p + D_i + D_q \tag{8}$$

where the elements in the right side are transmission, propagation, interface, and queuing delays. The interface delay is the delay on the end hosts (e.g. protocol overhead). Suppose the bandwidth is constant, and the length of server's statement is equal to length of hash value then the formulas for both models can be written as follows:

$$\begin{aligned} SAS:&\ \frac{3m+4h+2s}{B} + 3D_p + 3D_i + D_{q3} \\ Our\ Model:&\ \frac{5m+6s}{B} + 5D_p + 4D_i + D_{q5} \end{aligned} \tag{9}$$

Note the, B denotes the bandwidth of the network, and propagation and interface delays are functions of number of rounds. Subtraction of the previous two formulas gives us the following inequality:

$$2D_p + D_i + D_{q5} > \frac{2m-4h+4s}{B} + D_{q3} \tag{10}$$

If this inequality holds, then the network delay is larger in our model otherwise it is smaller. The extra amount of bits transmitted in our model is given which results in an extra transmission delay. However, in most situations transmission delay is not the dominate factor of the network delay [12].

Table 5 Objective comparison of our model with SAS.

	SAS model	Our model
Signature size	t	k
Signing cost	$3H + V + S$	$H + V + 3S + M$
Verifying cost	$V + 2H$	$V + S$
Storage requirement	$K + C + 3H$	K

H: Hash computation, *S*: traditional signing calculation, *V*: verification of signature, *t*: parameter for one-time signature ($t \cong 2k$), *K*: size of secret key or user's identity, *C*: size of signature counter, and *M*: mapping computation.

Table 6 Communication comparison of two models.

	SAS model	Our model
Number of rounds	3	5
Message length in round 1	$m + h$	m
Message length in round 2	$m + h + s$	$m + 3s$
Message length in round 3	$m + 2h + s$	$m + s$
Message length in round 4	–	$m + s$
Message length in round 5	–	$m + s$

m: Length of the shared key, *h*: length of random numbers and hash values, and *s*: length of signature and partial signature.

4.2. Security analysis

As demonstrated from the model description, the security issues of the proposed model are resolved as follows:

- It is infeasible for an adversary, including a client, to forge a signature on any message without the aid of a CA mounting adaptive chosen message attacks. Moreover, it is infeasible for a CA to frame a client without being held accounted. Intuitively, to forge a signature, an adversary may attempt to:
  (a) Forge a CA's half-signature: this attack is on the underlying public key signature scheme and as such is not specific to the proposed model.
  (b) Find the quantity *OIDi* : this attack essentially implies breaking one-way property of the underlying hash function.
- The client does not have to send a request to the bank containing client's certificate, since the SEM can specify the client by checking K_{client}. Furthermore, the model can protect the privacy of the clients from any eavesdropper, because one-time ID is used only once (i.e. more secure signature).
- Regarding DoS attacks in which it suffices for the adversary to food the CA server with well-formed request that will slowly grind to a halt. In the proposed model, it is appreciably more difficult for the attacker to launch this type of attack because of using one-time ID (CA will only issue a single signature operation per client for each session).
- To prevent man-in-the-middle attack, where an attacker can modify the contents of the request message, the model sends the message in an encrypted form. Unless an attacker obtains a K_{client} , an attacker cannot generate a legal request.
- With the proposed model, once the CA is notified of a certificate's revocation, the adversary is no longer able to interact with the CA to obtain signatures. Hence, potential compromise damage is severely reduced. In this instance, the existence of the signature implies that the client's certificate was valid at the time the signature was issued. This shows that the model significantly simplifies the semantics of signature verification.
- In the case, where an attacker compromise a CA and learns partial key, he can create a signature by colluding with client. Our model prevents this situation by using large K_{client}

key size that randomly changes after the services termination.

- Finally, when a number of clients are compromised (or a number of clients collude) there is no danger to other clients because each client is given his own modulus *n,* so if the private keys of the compromised clients will be exposed, the private key of all other clients will be unaffected (i.e. limited damage).

4.3. Security advantages

The proposed model provides a number of important security advantages over the previous scheme [8–10]:

- Our model provides deterministic guarantees for discovery of cheating by any party (cheating means, generating a signature and claming that the signature was generated by other parties) as oppose to the probabilistic guarantees of other approaches that can realize a reasonable low cheating probability only with a significant increase in the length of the public-key size.
- In the previous schemes, after seeing a valid signature produced by the proxy signer (CA in our case), the primary signer can swallow this signature. In our model, the proxy itself generates one-time private key of the signer and primary signer cannot learn it by any ways. As a result, swallow attacks do not pose any threat.
- The public-key based one-time protocol used in the key generation of the previous schemes brought back the dependency on the security of public-key algorithms, which have a long-term risk of being broken by the quantum computers. This dependency is not present in our scheme that can be implemented only using one-way function.
- Our model provides a more flexible dispute resolution process because parties can either prove their innocence or show that the other party has cheated without requiring involving the other party. This is in contrast to earlier work in whish the dispute regarding unforgeability against primary signer has to bring the proxy signer to court.

5. Conclusion

Banking transactions over Internet have become a common feature now. However, the question remains how secure these transactions. A solution to this has been proposed in this paper using the concepts of both IB-mRSA cryptosystem and one-time ID. The solution relies on partially trusted server for generating public/private keys for regular clients and has some interesting features such as built-in attack detection for clients and DoS resistance for server.

The major advantages of the presented model are: (1) Assist small limited–power devices in computing signatures; (2) Provide fast revocation of signing ability; (3) Limit damage from potential compromise; (4) Simplified signature verification; (5) Smaller signature lengths and faster verifying times. Finally (6) Verification transparency is another big advantage over previous verifiable-server approaches since there is no necessity to store past signatures to prove server's cheating. The only drawback of the proposed model is the increased number of rounds between the client and the server. However, decrease in the numbers of system's attacks is generally much more important than an increase in the bandwidth usage as far as communication delay is considered.

We implemented this model to experiment with it. Our implementation reveals that signature time is essentially unchanged from the client's perspective. Therefore, we believe that this architecture is appropriate for electronic banking system, where tight control of security capabilities is desirable. Our plans include migrating both of CA and of SEM to real-time platform. Extending the experimental performance evaluation to other platforms might also be very useful.

References

[1] J. Cleens, V. Dem, J. Vandewalle, On the security of today's online electronic banking systems, Journal of Computers & Security 21 (3) (2002) 257–269.

[2] Y.J. Yang, The Security of Electronic Banking, Technical Report, MD20783, University of Maryland, USA, 1998.

[3] N. Jin, F. Cheng, Network Security Risks in Online Banking, International Conference of Wireless Communication & Mobil Computing, Canada, 13–15 June 2005, pp. 1183–1188.

[4] K. Edge, R. Raines, M. Gremial, The use of attack and protection trees to analyze for an online banking system, in: Proceedings of the 40th Annual Hawaii International Conference on System Sciences, Hawaii, 3–6 January 2007.

[5] W. Stallings, Cryptography and Network Security Principles and Practice, third ed., Hall of India Private Limited, 2003.

[6] X. Ding, G. Tsudik, Simple identity based cryptography with mediated RSA, in: The Cryptographers Track RSA Conference, San Francisco, USA, 2003.

[7] D. Boneh, X. Ding, G. Tsudik, Identity-based encryption using mediated RSA, in: The 3rd Workshop on Information Security Applications, Korea, August 2002.

[8] S. Rajalakshmi, S. Srivatsa, Identity based encryption using mRSA in electronic transactions, Information Technology Journal 6 (3) (2007) 435–440.

[9] X. Ding, G. Tsudik, Experimenting with server-aided signatures, in: Proc. of the 9th annual symposium of network and distributed system, pp. 251–265, San Diego, USA, 2002.

[10] K. Satoshi, K. Imamoto, K. Sakurai, Enhancing Security of Security-mediated PKI by one-time ID, in: Proc. the 4th Annual PKI R&D Workshop, USA, April 19–21, 2005.

[11] M. O'Neill, Low-cost SHA-1 hash function architecture for RFID tags, in: Proc. of IEEE International Symposium on Circuits and System (ISCAS,07), USA, 27–30 May 2007, pp. 1839–1842.

[12] K. Bicakci, N. Baykal, Improved server assisted signatures, Computer Networks 47 (2005) 351–366.