Research Journal of Applied Sciences, Engineering and Technology 5(1): 270-274, 2013
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: June 02, 2012 Accepted: June 23, 2012 Published: January 01, 2013

Adaptive Universal Composability Framework for Server-Aided Threshold Signature

Xuan Hong and Luqun Li
Department of Computer Science and Technology, Shanghai Normal University,
Shanghai 200234, China

Abstract: The threshold signature scheme is a protocol that allows any subset of t parties out of n to generate a signature. Since the t members can cooperate together to compute the secret key, we introduce the server-aided threshold signature, which provides controllability for activating the signing function in a certain enhanced way. In this study, we present a server-aided threshold RSA signature protocol against the adaptive attacks. We give the universally composable secure model for the server-aided threshold signature primitive and prove that the proposed protocol is claimed to be well-formed, correct and unforgeability. As a separate contribution we also prove that it is also secure in the adaptive universal composability framework. After the discussion about the security and the performance, we claim that the protocol is practical and efficient.

Keywords: Adaptive security, server-aided, threshold signature, universal composability framework

INTRODUCTION

A threshold signature scheme is a protocol that allows any subset of t parties out of n to generate a signature. Meanwhile, it disallows the creation of a valid signature if fewer than t parties participate in the protocol. It should also be robust, meaning that corrupted parties should not be able to prevent uncorrupted parties from generating signature. Though threshold schemes based on the discrete logarithm problem are relatively straightforward to build, Basing threshold schemes on the RSA problem is more difficult, due to the fact that the modulus Φ (N) cannot be leaked to any of the shareholders. Boyd (1989) and Frankel (1989) present the first RSA based threshold signature independently. These earlier protocols additively share the signing key d among the parties. Rabin (1998) present the first reasonable efficient and robust solutions. However, they tend to be more complex and less efficient in comparison to the discrete logarithm schemes. These earlier protocols additively share the signing key d among the parties and they tend to be more complex and less efficient. Later, (Shoup, 2000) described his practical threshold signatures, which is widely regarded as the efficient threshold RSA signature scheme.

We consider a scenario that a classified file is stored in a file server which locates behind the company gateway. To access the file, someone outside the company should launch the successful authentication with a correct password between himself and the authentication gateway. However the file is so sensitive that the password is divided into several pieces which are shared among a group of members. Only at least t members together have the privilege to access the classified file. However, when some t members are corrupted, the file will be leaked out.

Therefore, we introduce the verifiable server, which is physical secure and can be trusted by the company gateway. The server-aided threshold signature scheme, where the signing function can be activated by a server, provides user controllability for activating the user's signing function. That is, if anyone has a power to activate the signing function of the servers, he can easily compute valid signatures for a specific organization without knowing the private key. Xu and Sandhu (2003) provided a general construction to build server assisted threshold signatures. Yang *et al*. (2006) also design two types of proactive secret sharing schemes which are suitable for server assisted threshold signatures.

In this study, we propose a server-aided threshold RSA signature protocol with adaptive universally composable security. In this protocol, the system is composed of one server and multiple users who share the private key together. The server and all users must cooperate to produce a valid signature. In addition, the proposed protocol has the adaptively UC-security, which makes the adversary forge signatures of previous periods more difficultly even though she can corrupt users at any time. The proposed protocol shares the secret with the additional sharing. The size of the signing key is constant and the size of the partial

Corresponding Author: Xuan Hong, Department of Computer Science and Technology, Shanghai Normal University, 100 Guilin Road, Shanghai 200234, China

signature is constant, which are independent to the number of the users. Furthermore, the threshold signature generation stage and the signature combining stage are completely non-interactive. At last, the correctness and security have been analyzed.

UNIVERSAL COMPOSABLE SECURE MODEL OF SERVER-AIDED THRESHOLD

Recently some efforts were made towards capturing the security requirements within the universal composability framework. This modeling has some significant advantages in designing and analyzing complex system. The framework maintains the security, regardless of execution environment. That is suitable for analyzing the mobile ad-hoc networks. Canetti introduced the universal composability framework (Canetti and Rabin, 2003; Canetti, 2004) as the new approach for designing and analyzing the security of cryptographic primitives and protocols. The main concern is to create new approach for the assessment of cryptographic protocol. It is guaranteed that a protocol with UC security maintains its security even when running concurrently with others. When analyzing multiprotocol systems, we analyze each protocol as if it is stand alone Then the composition theorem is used to deduce the security of all instances when running concurrently.

Security of the protocol π is achieved via comparing the protocol execution in the real-life model to the ideal model. This is formalized by considering an environment Z, which represents all the other protocols running in the system, including the protocols that provide inputs to and obtain outputs from. Security is required that comparing protocol π with an ideal functionality F, that is, whatever Z could achieve by attacking π, it could also achieve by interacting with F. in other words, if no environment can distinguish interactions with π from those with F, the protocol π is secure in the hybrid model. To make this precise, a special ITM S is introduced. The goal of S is to simulate the adversary's view of π, based on the information is willing to exchange with environment.

We define the functionality F_{SATSig} for the server-aided threshold signature, whose basic ideal is providing a registry service" where the users can register their (message, share) pairs. Any party that provides the right verification key can check whether a given pair is registered. It takes four types of inputs, which correspond to the four basic modules of the signature protocol: Setup, signature share, combine and verify.

We also deal with the case that the signer is corrupted. Corrupted signers are controlled by the adversary, which is different from the honest ones. If the signer is corrupted and F_{SATSig} is asked to verify (m, s), then F_{SATSig} allows the ideal-process adversary to force the answer to be "1", even if m was never before signed. We add the corrupted signer and make the whole framework more reasonable.

The functionality F_{SATSig} is also the standard corruption functionality, with an additional stipulation. When a user P is corrupted, F_{SATSig} records this fact and reports to the adversary all the signing and verification requests made by P. If P is the signer then the adversary gets also the signature share generation algorithm SS together with its current state. The verification algorithm V is deterministic guarantees consistency, namely that all verification requests for the same triple return the same answer. Verifying that V (m, σ) = 1 at signature generation time guarantees completeness, namely that if a signature was generated legally then it will pass verification. The check for forgery at verification time guarantees unforgeabilit*y*.

Proxy threshold signature F_{SATSig}:

- **Setup:** When received (Setup, *sid*) from party P, where *sid = (P, sid')*, F_{SATSig} sends (Setup, *sid*) to the adversary A. When received (Algorithms, *sid*, SS, SV, SC, V) from the adversary A, where SS is description of a PPT ITM and SV, SC, V are description of the deterministic PPT ITMs. F_{SATSig} sends (VeriAlgo, sid, SV) to the party P and (VeriAlgo, sid, V) to the server.
- **Signature share:** When received (ThSign, *sid*, *m*) from the party P, where *sid* = (P, sid'), F_{SATSig} sets $\sigma = SS(m)$, records the entry (m, σ), sends (ThSignature, sid, m, σ) to party P.
- **Combine:** When received (Combine, *sid*, m, σ_1, σ_2, …, σ_t) from the server S, F_{SATSig} check whether all the t shares are registered. If so, F_{SATSig} lets $\sigma = SC(\sigma_1,\ldots,\sigma_t)$, records the entry (m, σ), outputs (Combined, *sid,* m, σ) to S.
- **Verify:** When received (Verify, *sid*, m, σ) from party P, output (Verified, *sid*, m, σ, f)) to P, where f is determined as follows:
  - If V' = V and the entry (m, σ) is recorded, then set $f = 1$.
  - Else, if V' = V and no entry (m, σ) is recorded, then set $f = 0$.
  - Else, if V' $\neq$ V, then set $f = 0$.

SERVER-AIDED THRESHOLD SIGNATURE PROTOCOL

We present the server assisted threshold RSA signature protocol π, which realizes F_{SATSig} in a

straightforward way. All parties in the protocol run the following code:

Setup: It is run by the Key Generation Center (KGC). Given the security parameter *k*, the KGC perform RSA key generation with secure parameter *k* to obtain modulus *N*, where,

$$N = pq, p = 2p'+1, q = 2q'+1$$

with *p*, *q*, *p'*, *q'* themselves prime.
The KGC lets *M* = *p'q'*, computes the RSA components:

e,d, where $ed \equiv 1 \bmod M$

The KGC chooses the secret key *d'* at random for the server, then sets a_0 = *d-d'* and chooses a_i from { 0,…,m-l}, for $1 \le i \le n-1$. These numbers define the polynomial:

$$f(x) = \sum a_i x^i$$

The KGC computes the secret key for each signer S_i:

$$sk_i \equiv f(i) \bmod M$$

Now, the KGC computes the proofs for every signer S_i. Randomly select $v \in Q_N$ (subgroup of squares in Z_N). Sets:

$$v_i = v^{sk_i} \in Q_N$$

The signature share verification key is (v, v_i) and the public key is (*N*, *e*).

Signature share: We need a hash function $H(\cdot)$ mapping messages to elements of Z_N. If signer S_i wants to sign a valid signature on message *Mes*, it computes:

$$x = H(Mes)$$
$$\sigma_i = x^{2\Delta sk_i} \in Q_N, \text{ where } \Delta = n!$$

The signature share is σ_i and its proof is (z, c):

$$v' = v^r, x' = x^{4\Delta r}$$

$$c = H'(v, x^{4\Delta}, v_i, \sigma_i^2, v', x')$$

$$z = sk_i \cdot c + r$$

Combine: When the server receives the *t* signature shares from some *t* signers. It firstly checks whether these shares are valid:

$$c \overset{?}{=} H'(v, x^{4\Delta}, v_i, \sigma^2, v^z, x^{4\Delta z} \cdot \sigma^{-2c})$$

If the verification equations hold simultaneously, the server can compute the signature for the group as follows:

Defines $\lambda_{i,j} = \Delta \dfrac{\prod_{j' \in S \setminus \{j\}} (i - j')}{\prod_{j' \in S \setminus \{j\}} (j - j')}$

$$\delta' = \sigma_1^{2\lambda_{0,1}} \cdots \sigma_t^{2\lambda_{0,t}} = x^{4\Delta^2 d}$$

a, *b*, where $4\Delta^2 \cdot a + e \cdot b = 1$

The signature $\delta = \delta'^a \cdot x^b \cdot x^{d'} = H(Mes)^d$

Verify: Every party P can checks whether that:

$$\delta^e \equiv H(Mes) \bmod N$$

If so, the signature is valid.

SECURITY DISCUSSION

We will prove that for the server assisted threshold signature protocol is correct, consistent and unforgeable under chosen message attack against adaptive adversary. The proposed protocol also implements the signature functionality defined before.

Correctness: The following theorem shows that the signature produced by the proposed protocol satisfies the correctness.
Theorem 1: The proposed scheme is verifiable, if the server and all the users follow the issuing protocol.

Proof 1: Let S is the set of *t* or more users and their secret key is sk_i. The polynomial *f(x)* of degree *t-1* is expressed as:

$$f(x) = a_0 + a_1 x + \ldots + a_{t-1} x^{t-1} \bmod M_o$$

where,

$$f(0) = a_0 = d - d' \bmod M$$

Therefore, *f(0)* can be derived by the Lagrange formula:

$$\Delta \cdot f(0) \equiv \sum_{j \in S} \lambda_{0,j}^{S} \cdot f(j) \bmod M_o$$

On the other hand, from the proposed scheme, we learn:

$$\begin{aligned}\delta^e &= (x_{i_1}^{2\lambda_{0,i_1}^{S}} \cdots x_{i_t}^{2\lambda_{0,i_t}^{S}} \cdot x^{d'})^e \\ &= x^{4\Delta e \cdot (f(i_1)\cdot\lambda_{0,i_1}^{S} + \ldots + f(i_t)\cdot\lambda_{0,i_t}^{S} + d')} = x^{4\Delta e \cdot (f(0)+d')} \\ &= x^{4\Delta^2 \cdot H(Mes)} \bmod N_o\end{aligned}$$

If the signature δ is computed using the extended Euclidean algorithm, it must satisfy the verification equation. Hence, the theorem is proved.

Unforgeability: An outside adversary may try to derive a forged proxy signature. We will show that all attacks fail on our scheme.

Theorem 2: Suppose $H(\cdot)$, $H'(\cdot)$ be the random oracle, there is t or fewer corrupted proxy signers, the proposed scheme is a secure threshold proxy signature scheme assuming that the standard RSA signature scheme is secure.

Proof 2: If there is an adversary A can forge our proposed proxy signature, then there is an adversary B who can forge a RSA signature. We focus on how to simulate the adversary's view. Giving access to an RSA signing oracle which we use only when A asks for a signature share from an uncorrupted signer.

Let $i_1,\ldots,i_{t-1}$ be the set of corrupted proxy signers. B choose the k_i belonging to the set of corrupted signers at random form $\{0,\ldots, M-1\}$. We easily know that the statistical distance between the uniform distribution on $\{0,\ldots, M-1\}$ and the uniform distribution on $\{0,\ldots, M-1\}$ is $O(N^{1/2})$. Once all k_i are chosen, the values k_i for the uncorrupted signers are also completely determined, while can't be computed without M. B requests the RSA signing oracle to get δ. Then B easily computes:

$$x_t = x^{2\Delta \cdot k_{i_j}}$$

for the uncorrupted signer as:

$$x_t = y^{2(\lambda_{t,0}^{S} + e(\lambda_{t,i_1}^{S} \cdot k_{i_1} + \ldots + \lambda_{t,i_{t-1}}^{S} \cdot k_{i_{t-1}}))}$$

Considering the zero-knowledge simulation, B can also construct A's view without knowing the value k_t. This view includes the values of the random oracle $H'(\cdot)$ at those points where the adversary A has queried the oracle, so B is in complete charge of the random oracle. Whenever A asks a query to the random oracle, if the oracle has not been previously defined at the given point, B defines it to be a random value and in any case returns the value to the A.

When an uncorrupted proxy signers is supposed to generate a proof of correctness, B choose:

$$c \in \{0,\ldots,2^k - 1\},\ z \in \{0,\ldots,2^{|N|+2k} - 1\}$$

at random and for given values x_i, x, defines the value of the random oracle at $(v, \tilde{x}, v_t, x_t, v^z v_t^{-c}, \tilde{x}^z x_t^{-2c})$ to be c. Since B has not defined the random oracle at this point before, it is free to do so now. The proof is just (z, c). It is straightforward to verify that the distribution produced by B is statistically close to perfect.

Hence, B can outputs a valid RSA signature (x, y), while x has not been queried, with negligible probability. The scheme is proved.

Active security: As in the passive case, the simulation is statistically close to the protocol until the user is corrupted. In the active case, the adversary should corrupt any user at any time. The simulation should forge the valid signature and parse the corrupt signer's signature share verification. As the functionality F_{SATSig} store all the signature share verification algorithms in his database, he can answer all the queries. Since F_{SATSig} can answer the corrupted user's secret key share queries in the simulation, F_{SATSig} can correctly answer A's signature share verification queries. Thus, security of the proposed protocol can be reduced to the security of RSA problem.

Lemma 3: The proposed protocol is well formed, correct, unforgeable relative to the environments which corrupt and adaptively, at most *t-1* parties.

Theorem 4: The proposed protocol securely realizes the ideal functionality of server-aided threshold signature.

Briefly, this result is shown by constructing a UC simulator S, which will generate on its own a set of keys for the signature scheme by executing internally an instance of π. Let A be an adversary that interacts with parties running π in the ideal model. We construct an ideal process adversary (simulator) S such that the view of any environment Z of an interaction with A and π is distributed identically to its view of an interaction with S in the ideal process of F_{SATSig}. Generally

speaking, simulator S runs an internal copy of A and each of the involved parties.

Thus using the private keys, it can trivially simulate the environment Z's view of π by simply following the protocol to generate signature. We can observe that the only way it could differ from actual execution is if Z can produce a valid signature that was not legally generated. However, the unsolvable of the RSA assumption ensures that such event occur with negligible probability.

PERFORMANCE DISCUSSION

The study shares the secret key by using a simple Lagrange formula but not extra cryptographic techniques. We hide the secret information *M* with the subgroup of squares. The proposed protocol satisfies the server assisted property, when the user wants to enjoy a threshold protection of her private signing function by taking advantage of multiple servers. It shares the secret with the additional sharing. The size of the signing key is constant and the size of the signature is constant, those are independent to the number of users. Furthermore, the signature generation stage and the signature combining stage are completely non-interactive. Owing to its simple algorithm and fewer parameter requirements needed, the proposed protocol requires the fewer calculations and the fewer transactions.

Mobile agent is one of the best application areas of server-aided threshold signature scheme. The mobile agent is autonomous software entity, which migrates across different execution environments through the network. The characteristics of the mobile agent, mobility and autonomy, make it ideal for electronic commerce applications in many ways, such as ubiquitous computing, mobile commerce. As the agents are not trustworthy, or may be malicious, there are great challenges. One way to reduce the computation cost on mobile devices is to get help from a verifiable and a powerful server.

A possible solution is to adopt the server assisted threshold signature. We store securely secret private keys by servers instead of each mobile agent and perform securely their cryptographic functions.

CONCLUSION

In this study, we present a simple, efficient and adaptively secure server-aided threshold RSA signature protocol. We give the universally composable secure model for the server-aided threshold signature and prove that the proposed protocol is claimed to be well-formed correct, consistent and *t-1* unforgeability. As a separate contribution we also prove that it is also secure in the universal composability framework. It shares the secret with the additional sharing. The size of the signing key is constant and the size of the signature is constant, those are independent to the number of users. The proposed scheme reduces large amounts of modular exponential computations and communications. It can be applied to the mobile agent system, regardless what the environment it interacts with.

ACKNOWLEDGMENT

This Study is supported by the National Natural Science Foundation of China (NSFC) under Grant No. 61003215 and Innovation Program of Shanghai Municipal Education Commission No.12YZ072.

REFERENCES

Boyd, C., 1989. Digital Multisignatures. Cryptography and Coding, (Ed.), Oxford University Press, Oxford, pp: 241-246.

Canetti, R. and T. Rabin, 2003. Universal composition with joint state: In Crypto. LNCS, 2729: 265-281.

Canetti, R., 2004. Universally Composable Signature, certification and authentication. Proceedings of the 17th IEEE workshop on Computer Security Foundations, IEEE Computer Society Press, New York, pp: 219-233.

Frankel, Y., 1989. A practical protocol for large group oriented networks: In EUROCRYPT. LNCS, 434: 56-61.

Rabin, T., 1998. A simplified approach to threshold and proactive RSA: In CRYPTO. LNCS, 1462: 89-104.

Shoup, V., 2000. Practical threshold signatures: In EUROCRYPT. LNCS, 1807: 207-220.

Xu, S.H. and R. Sandhu, 2003. Two efficient and provably secure schemes for server-assisted threshold signature: In CT-RSA. LNCS, 2612: 355-373.

Yang, J.P., K.H. Rhee and K. Sakurai, 2006. A Proactive Secret Sharing for Server Assisted Threshold Signatures: In HPCC. LNCS, 4208: 250-259.