Providing High Availability to Time-Stamping Services Using Threshold Signature

Joseph K. Liu[1], **Karyin Fung**[1], **Duncan S. Wong**[2]
[1]Department of Information Engineering,
The Chinese University of Hong Kong, Hong Kong.

[2]Department of Computer Science,
City University of Hong Kong, Hong Kong.

Abstract. We present a solution to the high availability issue that arises when digital time-stamping authority tries providing robust services to its users. High availability of time-stamping services is very important because any suspension of service due to system failure or potential attacks results in serious monetary losses. In this paper, we present a new approach that deploys an RSA threshold signature scheme to offer high availability for multiple time-stamping servers. It allows the sharing of signing key into *n* servers while the robustness of the system is unaffected even if some subsets of less than t servers are corrupted and work together. Our proposed solution provides a strong prevention measure instead of recovery measure of physical operation and cryptographical attack. Thus, it enjoys a higher level of security protection than traditional high availability protocol.

Keywords: Time-stamping, High Availability

1 Introduction

Time-stamping [3] is an online notary mechanism which certifies data that has existed and has not been altered since a specific point in time. One challenge of such a system is that any delay on the system could make undeterminable suffering to other involved parties. Such delay can be caused by system failure, denial-of-service attacks or compromise of server secret key. For example, if the server of the Time-Stamping Authority (TSA) is being attacked and cannot provide normal service, a large number of users can neither request time-stamping service nor get verification. Furthermore, if the server is corrupted, the time-stamps it has issued become invalid due to the compromise of the secret key. Meanwhile it looks unfair at the point-of-view of an user who has no control of the operation of the TSA.

In order to provide a trust-worthy and robust service, high availability of the TSA is necessary. [1] describes a protocol for improving the availability of both the hash-and-sign and the linkage-based time-stamping services. The main idea is to send the time stamp requests to all servers and obtains time stamps from all of them except those being inaccessible at the moment. [1] also addresses three main events of high availability issues of time-stamping service, namely (1) Broken Cryptography and compromised keys; (2) Service unavailability and (3) Loss of servers data. We mainly focus on the first one as the last two seem to be regarded as physical security. The first one is also the main concern of all security service providers, such as Certificate Authority (CA) and Time-Stamping Authority (TSA).

In this paper, we give a novel approach to provide high availability using threshold signature. It enjoys a higher level of security protection by providing a strong prevention measure of cryptographical attack. Threshold signature [5] allows a group of people to hold the key together such that a subset of them is enough to produce valid signature. We refer such a scheme as a *(t, n)*-threshold signature scheme if there are n players given and a subset of at least *t+1* players is enough to give a valid signature on message m. The system is said to be robust if and only if there are at least $t+1$ players correctly compute their corresponding parts of the signature. It is information theoretically infeasible to compute a valid signature for a subset up to at most *t* players know the secret.

This paper is organized as follows: In Sec. 2, we outline the basic idea of time-stamping and the objective of providing high availability of time-stamping services. In Sec. 3, we present our proposed protocol. Security analysis will be given in Sec. 4 and conclusion is followed in Sec. 5.

2 Background

Time-Stamping The user sends a time-stamp request to the TSA. The time-stamp request contains the hash of the original document or the transaction information. The TSA appends the current time *t* to the hash *h* and uses its private key to sign it and produce a signature $s = sign_{TSA}(t, h)$. The signature is sent back to the user with the time-stamp. There are several weaknesses of this scheme. If the signing key of the TSA is compromised, the old time-stamps became unreliable because it is impossible to verify whether the time-stamp was issued before the leakage of the signing key or after the leakage.

In a Linear Linking Scheme (LLS) [3], the linking information (for example, the hash of the previous time-stamp issued by the TSA) is added into the current time-stamp for linking all the time-stamps together and form a chain. This preserves the correct order of time-stamps. Yet the verification process could be costly as it has to verify the whole chain. Tree-Like Schemes [4] reduces the complexity and recent linking protocols [2] using authenticated graph can

achieve optimal complexity.

High Availability In the service level, availability means expected periods of service available and acceptable downtime. It is measured using the period of time when applications are available during the time they are expected to be available. High availability is the minimization of planned and unplanned outage incidents and outage downtime. Many sites are willing to absorb a short period of downtime rather than pay the much higher cost of providing fault tolerance. Companies use high availability systems for applications that must be restored quickly but can withstand a short interruption in the event of a failure or an operator error. It is especially concerned in a system providing security related services, such as certificate authority (CA) or time-stamping authority (TSA).

In providing time-stamping service, high availability is more important than certificate authorization service because it is not allowed having any delay in time. Replying parties do not depend on any other parties but only on the TSA which is assumed trustworthy and accurate. Furthermore, mechanism is needed to protect the authenticity of time-stamps when the signing key is compromised. Besides, robustness of the TSA should be protected even the signing key is compromised in order to provide high availability.

Threshold Signature Scheme Below is a brief review of a practical threshold signature scheme proposed by [7]. It is based on RSA and has some desired features such as unforgeable and robust; non-interactive in share generation and verification, and constant size of signature share. The basic idea is to distribute the signing key to l players such that $l \geq 2t+1$ where t is the threshold value, the maximum number of corrupted players. Let k be another threshold which is the minimum required number of authorized players such that $k \geq t+1$. The details are as follows.

The Dealer. The dealer chooses two large equal-length primes randomly, $p = 2p'+1$ and $q = 2q'+1$ such that p' and q' are prime. The RSA modulus is $n = pq$. Let $m = p'q'$ and e be the RSA public exponent which is a prime larger than l. The public key is (n, e). The private key is d such that $de \equiv 1 \bmod m$. The dealer sets $a_0 = d$ and choose a_i at random from $0, \ldots, m-1$ for $1 \leq i \leq k-1$. The number $a_0, \ldots, a_{k-1}$ define the polynomial

$$f(X) = \sum_{i=0}^{k-1} a_i X^i \in Z[X]$$

For $1 \leq i \leq l$, the dealer computes $s_i = f(i) \bmod m$. This number s_i is the secret key share SK_i of player i. The dealer chooses a random number $v \in Q_n$ where Q_n is the subground of squares in Z_n^*. For $1 \leq i \leq l$, the dealer computes $v_i = v^{s_i} \in Q_n$. These elements form the verification keys: $VK = v$ and $VK_i = v_i$. Let $\Delta = l!$. For any subset S of k points in $\{0, \ldots, l\}$, and for any $i \in \{0, \ldots, l\} \backslash S$ and $j \in S$, let

$$\lambda_{i,j}^s = \frac{\prod_{f \in S \backslash \{j\}} i - j'}{\prod_{f \in S \backslash \{j\}} j - j'} \in Z$$

From the Lagrange interpolation formula, we have,

$$\Delta f(i) \equiv \sum_{j \in S} \lambda_{i,j}^S f(j) \bmod m$$

Generating a Signature Share. Let $x = H(M)$ where H is some hash function and M is some message. The signature share of player i : $x_i = x_{2\Delta s_i} \in Q_n$ along with the "proof of correctness": $\widetilde{x} = x^{4\Delta}$. Let $L(n)$ be the bit-length of n and H' be another hash function which outputs L_1-bit integers. Player i chooses a random number $r \in \{0, \ldots, 2^{L(n)+2L_1} - 1\}$, and computes $v' = v^r$, $x' = \widetilde{x}^r$, $c = H'(v, \widetilde{x}, v_i, x_i^2, v', x')$, and $z = s_i c + r$. The proof of correctness is (z, c) which is verified by $c = H'(v, \widetilde{x}, v_i, x_i^2, v^z v_i^{-c}, \widetilde{x}^z x_i^{-2c})$.

Combining Shares. Given the valid shares of players $S = \{i_1, \ldots, i_k\} \subset \{1, \ldots, l\}$. Let $x_{i_1}^2 = x^{4\Delta S_{i_j}}$. To combine the shares, compute $w = x_{i_1}^{2\lambda_{0,i_1}^2} \ldots x_{i_k}^{2\lambda_{0,i_k}^S}$. Then compute $e' = 4\Delta^2$ and the combined signature y such that $y^e = x : y = w^a x^b$ where a and b are integers such that $e'a + eb = 1$ which can be obtained using extended Euclidean algorithm on e' and e.

3 Our Proposed Protocol

Our protocol is motivated by [7]. We first give a high availability model of time-stamping service using absolute time-stamping, then the model of linkage-based time-stamping.

3.1 Absolute Time-Stamping

In order to provide high availability, using multiple servers is necessary. In our model, there is a gateway which is the only machine opened for public to accept time stamp requests. An overview of our model is shown in Fig. 1.

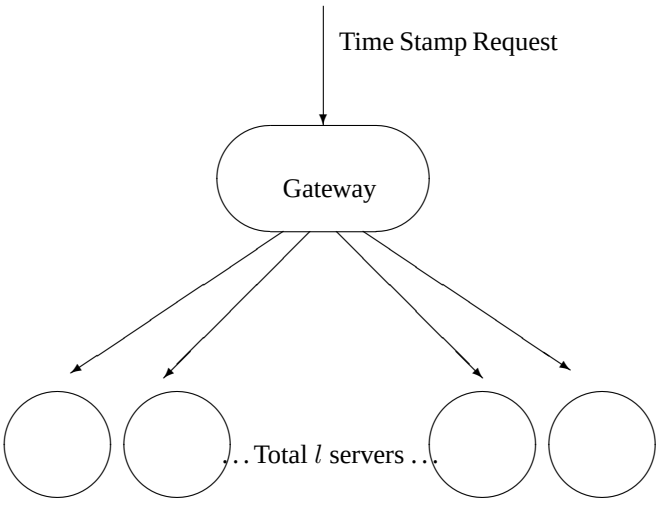

Figure 1

Preparation. Suppose there are l servers, all connected to the centralized gateway which is also connected to the time

source (for example, a GPS clock). The administrator prepares the following: Two large equal-length primes p and q such that $p = 2p' + 1$ and $q = 2q' + 1$ with p' and q' are also prime numbers. Computer $n = pq$ and $m = p'q'$. Choose a RSA public exponent e such that e is prime and $e \geq l$. Publish (n, e) as the RSA public key. Compute d such that $deequiv1 \bmod m$. Define the polynomial such that $a_0 = d$. Compute the secret key shares for $1 \leq i \leq l$. Distribute s_i to each of the servers (we assume the distribution channel is a secure channel). Choose a random number v and computer the verification key $v_i = v^{s_1} \in Q_n$ where Q_n is the subgroup of squares in Z_n^* for $1 \leq i \leq l$. Then he sends v and all the v_i to the central gateway for the purpose of verifying signature shares.

Signing Time Stamp. When there is a time stamp request, the gateway randomly chooses k servers ($2k - 1 \geq l$) for signing. Let h is the hash of the document or transaction information that is going to be time stamped and T be the time the gateway receives the time stamp request. Let $x = H(h, T)$ where H is an one-way hashing function. Each of the i servers signs it using standard RSA signature generation.

Combining Signature Share. Let S be the subset of k servers which are chosen by the gateway to give the signature shares. For any $i \in \{0, \ldots, l\} \backslash S$ and $j \in S$, the gateway compute: $\lambda_{i,j}^S = \Delta \frac{\prod_{f \in S \backslash \{j\}} (i - j')}{\prod_{f \in S \backslash \{f\}} (j - j')} \in Z$ where $\Delta = l!$, $w = x_i^{2\lambda_{0,i_1}^S} \ldots x_i^{2\lambda_{0,i_k}^S}$ where λ's are the integers defined above, and $e' = 4\lambda_2$. Find two integers a and b using extended Euclidean algorithm such that $e'a + eb = 1$. The combined signature y is computed as $y = w^a x^b$.

3.2 Linkage-based Time-Stamping

Our proposed protocol is compatible to linkage-based time-stamping. Hashing is the main cryptographic tool deployed in linkage time-stamping protocol. In our protocol, we split the hashing process and the signing process into two parts. Hashing is done in the central gateway and signing is done by multiple servers using threshold signature in order to spread the secret key. If linkage protocol is used, only the linkage information needs to be added to the input of the hashing function in the central gateway.

4 Security Analysis

Theorem 1 *Under the assumption that the standard RSA signature scheme is secure, the system remains secure even in the presence of t corrupted servers where the total number of servers is $l \geq 2t + 1$.*

Proof. Define a polynomial $f(x)$ of degree t such that $f(0) = d$ which is the signing secret of the system. As there are $t + 1$ points defined in the polynomial $f(x)$ (corresponding to $t + 1$ corrupted servers) we can compute its coefficients including the constant term d. Less than $t + 1$ points is information theoretic infeasible to restore any coefficient (corresponding to up to a maximum of t corrupt servers are unable to compute the signing secret.

Theorem 2 *With regard to the "proof of correctness", one can get soundness and statistical zero-knowledge.*
Proof. It follows the proof in [7] and is skipped here.

Furthermore, the central gateway knows nothing about the secret of each server. This is observed from the combination of signatures that the combiner only needs to know the public verification keys of each server together with the proof of correctness.

Follow from above, we can see that the security of the system can be regarded as a higher level since the central gateway is the only machine which is opened to public. Yet it does not contain any secret information. Those machines containing secret information cannot be accessed by public. This is a highly secure scenario.

5 Conclusion

This paper addresses a new approach to the high availability of time-stamping service using threshold signature. Assume there are l servers. The system remains unaffected even if there are up to t servers corrupted or out of service provided that $l \geq 2t + 1$. Unlike [1], the timestamp issued before does not need to take any renewal process even if at most the signing keys of t servers are compromised. In addition, users only need to send one timestamp request instead of l timestamp requests. The high availability protocol is transparent to users.

References

[1] A. Ansper, A. Buldas, M. Saarepera and J. Willemson. Improving the availability of time-stamping services, ACISP 2001, pp. 360–375, 2001. LNCS 2119

[2] A. Buldas, P. Laud, H. Lipmaa and J. Villemson. Time-Stamping with Binary Linking Schemes, CRYPTO 98, pp. 486–501, 1998. LNCS 1462.

[3] S. Haber and W. Stornetta. How to Time-Stamp a Digital Document. J. Cryptology, 3(2):99-111, 1991.

[4] S. Haber and W. Stornetta. Secure Names for Bit-Strings. Proc. of the 4th ACM Conference on Computer and Communications Security, pp. 28–35, 1997.

[5] T. Rabin and M. Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority. STOC 1989, pp. 73-85, 1989.

[6] A. Shamir. How to Share a Secret. Communications of the ACM, 22(11): 612-613, 1979.

[7] V. Shoup: Practical Threshold Signatures. EUROCRYPT 2000, pp. 207–220, 2000. LNCS 1807