ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2015

Survey of One Time Signature Schemes on Cloud Computing

Revatthy Krishnamurthy[1], K.P. Kaliyamurthie[2]

[1]M.Tech Scholar, Department of Computer Science & Engineering, Bharath University, Chennai, India

[2]Professor, Department of Computer Science & Engineering, Bharath University, Chennai, India.

ABSTRACT : Cloud computing is a technology and large - scale computing resources to effectively integrate, and the resources are computed based on cryptographic secure hash functions. The biggest problem of one time signature scheme is the key management. An efficient key management is needed to make one-time signature scheme and the Merkle signature schemes feasible. This paper presents detailed study on one-time signature and Merkle signature schemes.

KEYWORDS*:* Cloud computing, one-time signature scheme, Merkle signature scheme, Key generation.

I. INTRODUCTION

Cloud is a large group of interconnected computers, which is a major change in how we store information and run application. Cloud computing is used for many bigdata applications and it is cost effective. Data storage and sharing services in the cloud with three entities such as the cloud, the third party Auditor (TPA), and users who participate as a group includes one original user and a number of group users.

The original user is an original owner of data, and shares data in the cloud with other users [2]. A single message using a given piece of private or public information. The conventional signature schemes like RSA, the same key pair can be used to authenticate large number of documents. One-time signatures by Merklecalled Merklesignature scheme, which does not require new key pairs for each message. Simple digital signature scheme for fixed - length messages using a one-way function is Lamport's one-time signature scheme. Diffie OTS [3], the merkle OTS [4], the Winternitz OTS [4,5], etc are different signature schemes. The security of one-time signatures is based on cryptographic secure hash functions. The Lamport one-time signature scheme is a signature scheme in which the public key can only be used to sign a single message. The Winternitz one-time signature scheme, the signature size can be reduced at the cost of hash operations of one-time signature scheme is an efficient key management that reduces the amount of public keys and their size is needed. The Merklesignature scheme, in which one public key is used to sign many messages [6][10].

The organization of the paper is as follows. Section II presents the review of related work. One-time signature schemes are explained in section III. Merkle signature scheme is discussed in section IV and section V concludes the paper.

II. REVIEW OF RELATED WORK

Cloud computing provides services to customers.Private, community, public and hybrid are the four models of cloud computing [7] [8] [9]. One - time signatures are efficient and secure. Typically, signature parameters are initialized well ahead of the time when messages are to be signed and verified. Several schemes were proposed that use classical authentication schemes such as digital signatures RSA.[EIGamal] for group-based transformations. However, these conventional methods typically have a high computational costs, and regard to the efficiency of the emerging applications. In contrast, one-time signatures provide the required security services with less computation overhead.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2015

III. ONE-TIME SIGNATURE SCHEMES

In one-time signature scheme[10], we must know in advance how many signatures the user will ever plan on sending. In this scheme, we have to generate on pairs (PK_i, SK_i) and build a tree using a collision - resistant hash function, h, by hashing each pair of adjacent nodes recursively up the tree to the root. The root is the public key, r.

SK = $\{(Pk_i, Sk_i)\}_{i=1,....n}$

The tree will be built by n pairs of public keys and secret keys.

The next level is $S_1^1 = h(Pk_1, Pk_2), S_2^1 = h(Pk_3, Pk_4)..... S_{n/2}^1 = h(PK_{n-1}), PK_n$

The second level is constructed as follows:

$S_1^2 = h(S_1', S_2'), S_2^2 = h$ (S 1/3, S¼), ... $S_{n/4}^2 = h(S_{n/2}^1 - 1, S_{n/2}^1)$

A tree is built with one-time signature schemes (PK, SK)s at the nodes. The tree grows from the top to bottom and expensive than the one-time signature scheme. Every time a new pair generated is an expensive task of generating new pairs.

1. *Lamport one - Time signature scheme*

The security of Lamport signature is based on cryptographic hash function. Here, the public key is used to sign a single message. Other secure hash functions also can be used because this signature scheme is very adjustable. If a hash function becomes insecure, it can be easily exchanged by another secure function. The key generation,signing algorithm and verification algorithm are described as follows[10].

Key generation
Consider a hash function
H : $(0, 1)^*$
H : $(0, 1)^s$ be a cryptographic hash function.
To sign a message M = $(0, 1)^k$ and choose 2 * k random numbers X_{ij} with $1 \le i \le k$ and j = $\{0,1\}$.

For each i and j computeY_{ij}= H (X_{ij}).

Here $Y_{ij}$are the public by key and the $Y_{ij}$are the private key values for each 2 * k values.

Signing a message

Given a message M = m_1, m_2, ...,mk with $m_i \sum \{0,1\}$and the private keys X_{ij} with 1≤ i ≤ k and j = $\{0,1\}$ .We have to check mi equals to 0 or 1 for each i. If it is 0, then $sig_i = X_{i0;}$otherwise $sig_i = X_{i1}$

The signature is the concatenation of all $sig_i$for i = $\{1,.....k\}$. Therefore, sig is evaluated as sig=(sig_1//sig_2//.....//sig_k) with// denotes the concatenation operator.

Signature verification

For a given message M = m1, m2, ...mk with mi $\sum \{0,1\}$and the signature of the given message is sig = (sig_1 // sig_2 // ..... // sig_k) and $Y_{ij}$is the corresponding public key of the Lamport One-Time signature scheme.For each 1≤ i ≤ k the hash value H (sig_i) gets computed. If on m_i = 0 then H (sig_i) must be H (sig_i) = Y_{i0}otherwise. H (sig_i) must be H (sig_i) = $Y_{ij}$to be a valid signature. The Loss scheme is the big size of the public and private key.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2015

2. *Winternitz One-time signature scheme*

TheWinternitzone-time signature scheme, the signature size can be reduced at the cost of hash operations[10].Consider a message M = $\{0,1\}^k$, 2* k hash values to be saved. A hash function must have atleast 160 bits to achieve a security at least O (2^{80}). Therefore, the private and public key must have at least 160 * 2 * k bits.This results in a total size of the public key and private key of 160 * 2 * 160bits = 51200 bits = 6400 bytes. A public key the LOTSS is 50 times larger than an equivalent 1024 - bits RSA public key. The signature sig is the concatenation of k hash values[10].

Key Generation

Let the hash function H : $\{0,1\}^* \rightarrow H : \{0,1\}^s$ be a crytographic hash function. The time is calculated as t = [s/w] + [($\log_2$ [s/w]) + 1 + w/w]. At first the parameters w, with w Σ N is chosen and then the above time is calculated. Private key is generated by computing X_i by choosing random numbers i.eX_1, ...X_t $\sum \{0,1\}$ [10].

The public key Y is generated in the next step by computing

$Y_i = H^{2^w} - 1$ (x_i) for i = 1, ... t

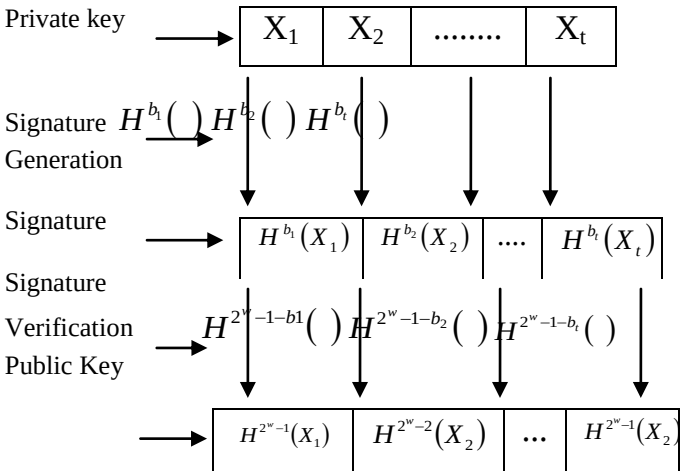

Figure 1.1: Signature generation and verification with the WinternitzOne-time signature Scheme.

The public key Y = H (Y_1 // ..... // Y_t) is the hash value of the concatenation of all Y_i with i = 1, ... t.

Signature Generation

Let M be the message and M = m_1, ...m_s $\sum \{0,1\}$ be the message to be signed. X_1, ...$X_t$the private key. Now, the message M is split up into [s/w] blocks b_1....., b [s/w] of the length w. Now treat b_i as the integer encoded by the respective block and compute the checksum.

[s/w]

C = $\sum 2^w - b_i$.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2015

We then split the binary representation i = 1of length w. Let us take b_i as the integer encoded by the block b_i and compute $sig_i = H^{bi}(x_i)$ for i = 1,..... t with $H^o(x_i) = (x_i)$. The signature sig = (sig_1//..... // sig_t) of the message M is the concatenation of all sig_i for i = 1, ...t.

Signature verification

To verify a signature sig = (sig_1//..... // sig_t) for a given message M = $\{0,1\}^s$ the parameters b_1, ...b_t are computed first.

for i = 1, ... , t $sig_i^1 = H^{2^w-1-bi}(sig_i)$ is calculated.

$sig_i' = H^{2^w-1-bi}(sig_i) = H^{2^w-1-bi}(H^{bi}(Xi)) = H^{2^w-1-bi}(sig_i)$ is calculated.

$$sig_i' = H^{2^w-1-bi}(sig_i) = H^{2^w-1-bi}(H^{bi}(Xi)) = H^{2^w-1}(X_i) = H^{2^w-1}(X_i) = Y_i$$

Hence if $Y' = H(sig_1^i // \ldots\ldots // sig_t')$ equals

Y = H (Y_1//..... // Y_t) the signature is valid. Or else signature is invalid.

A bigger parameter w will result in smaller signature size, but the signature generation time and the signature verification time will increase. So, choosing a too big value for w is not recommended.

IV. MERKLE - SIGNATURE SCHEME

Key management is the biggest problem in one-time signature scheme. Public key exchanging is very complex and it has not been modified and should be rather short. A new public key is used for every signature and it is quite big in One-time signature scheme. Using an efficient key management, the amount and size of public key used is reduced to make one-time signature scheme feasible. Merkle introduced the Merkle signature Scheme (MSS), in which one public key is used to sign many messages[10].

Key generation

The root of the tree $a_{n,o}$ is the public key pub of the Merkle signature scheme. The Merkle Signature Scheme can only be used to sign a limited number of messages with one public key pub. The possible number of messages as N = 2^n. The first step of generating the public key pub is to generate the public keys X_i and private keys Y_i of 2^n one-time signatures for each public key Y_i, with $1 \leq i \leq 2^n$, a hash value hi = H (Y_i) is computed. With these hash values h_i a Merkle tree is build. The node of the tree $a_{i,j}$, where i denotes the level of the node. The distance between the leaf and node is the level of the node. Therefore, level i=0 is the leaf of the tree and level i=n is the root of the tree.We number all nodes of one level from the left to right, so that $a_{i,o}$is the left most node of level i. In the Merkle tree, the hash values hi = $a_{i,o}$. Each inner node of the tree is the hash value of the concatenation of its two children. So,

$a_{i,o}$= H ($a_{o,o}$// $a_{o,1}$) and $a_{2,o}$ = H ($a_{1,o}$ // $a_{1,1}$).

Therefore, a Merkle tree with 2^n leaves and 2^{n+1}- l nodes is build[10].

Signature generation

The Merkle signature scheme, the message M is signed with a one-time signature scheme, resulting in a signature sig^1.$sig^1$is evaluated by using one of the public and private key pairs (Xi, Yi). The corresponding leaf of the hash tree to a one-time public key Yi is $a_{o,i}$= H (Yi). The path of root is A. The path A consists of n+1 nodes i.eAo, ... An, with Ao = $a_{o,i}$= pub being the leaf and A_n = $a_{n,0}$ = pub being root of the tree. To compute this path A, we need every child of the nodes A_1, ...A_n is a child of A_i + 1.The next node is calculated by a brother node called as $auth_i$, so that

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2015

A_{i+1} = H (A_i // $auth_i$). Hence, to compute every node of the path A, we need brother nodes. These nodes, plus the one-time signature sig' of M is the signature sig = (sig' // $auth_2$ // $auth_3$ // .... $auth_{n-1}$) of the Merkle Signature Scheme[10].

Signature Verification

When the receiver verifies the signature of the message i.esig' ,once if it is valid. Then Ao = H (Y_i) is calculated and verified by the receiver. Here, Ao = H (Y_i) is hash value of the public key of one-time signature.

V. CONCLUSION

One-time signature schemes are effective methods, used for key generation and signature verification. The biggest complexity of one-time signature scheme is the key management. The Merkle signature scheme which is used to overcome the problem of reducing the amount and size of the public key and is used to sign many messages. This paper deals detailed study about the importance of one-time signature scheme and Merklesignature scheme on cloud computing.

REFERENCES

[1] P. Mell and T. Grance, "Draft NIST working definition of cloud computing".
[2]. Nupoor et al., "Third Party Auditing (TPA) for data storage security in cloud with RC 5 Algorithm", 3(11), pp.1032-1037, 2013.
[3] Leslie Lamport, "Constructing digital signatures from a one way function". Technical Report SRI-CSL-98, SRI International Computer Science Laboratory, 1979.
[4] Ralph C. Merkle. A certified digital signature.In Gilles Brassard, editor. CRYPTO, vol 435 of LNCS, pages 218-238, Springer, 1989.
[5] Chris Dods, Nigel Smart, and MartijnStam, "Hash based digital signature schemes in Cryptography and coding", pages 96-115, Springer, 2005.
[6] Georg Becker, "Merkle signature schemes, Merkle Trees and their Cryptanalysis", pp. 1-24, 2008.
[7] Heiser J. (2009) what you need to know about cloud computing security and compliance,Gartner, Research, ID Number : G00168345.
[8] Amzon Elastic compute cloud (ECZ). <http://aws.amazon.com/secuirty Accessed: [January 2013].
[9] Choudary V (2007). Software as a service: implications for investment in software development in: International conference on system sciences, pp. 209,2007.
[10]"https://www.emsec.rub.de/media/crypto/attachments/files/2011/04/becker_1.pdf".