Available online at www.sciencedirect.com
SCIENCE DIRECT®
APPLIED MATHEMATICS AND COMPUTATION
ELSEVIER Applied Mathematics and Computation 167 (2005) 118–124
www.elsevier.com/locate/amc

Improving Lamport one-time signature scheme

Ming-Hsin Chang *, Yi-Shiung Yeh

Institute of Computer Science and Information Engineering, National Chiao-Tung University 1001, Ta-Hsueh Road, Hsinchu 300, Taiwan, ROC

Abstract

One-time signature scheme, a kind of digital signature schemes, is used to sign at most one message; otherwise the signature can be forged. One of the advantages is that the one-time signature generation and verification are very efficient and it is useful for chip cards, where low computation complexity is required. Lamport first invented a one-time digital signature scheme based on one-way functions. However, the Lamport one-time scheme requires a large amount of space for storage of authentic information if a large number of messages are signed. In this paper, we improve the Lamport one-time signature on the amount of storage space for public keys and signed message saving storage space and propose an efficient scheme to sign a long message. Thus, we make the Lamport one-time signature useful in practice.

Keywords: One-time signature; One-way hashing function; Encode; Decode

* Corresponding author.
E-mail address: ucc@cht.com.tw (M.-H. Chang).

0096-3003/$ - see front matter © 2004 Elsevier Inc. All rights reserved.
doi:10.1016/j.amc.2004.06.108

1. Introduction

One-time signature schemes were first proposed by Rabin [5] and Lamport [2] and based on the idea of committing public keys to secure keys using one-way functions. For more 25 years, Lamport one-time signature schemes have been proposed and investigated by many researchers [1,4,8,9]. Indeed, one-time signature schemes have found many interesting applications, including on-line/off-line signatures, digital signatures with forward security properties, broadcast authentication protocols and proxy signatures [7], etc.

In recent years, one-time signature schemes have attracted more and more attention, as an attractive alternative to the traditional signature schemes based on public key cryptography. One of the main advantages of one-time signature schemes is their reliance on one-way functions that can be implemented using fast hash functions such that SHA-2 [3] or MD5 [6]. The resulting signatures are the order of magnitude faster than signatures based on public cryptography applying on the resource-constrained, small devices, such as cell phones, pagers, smart cards etc. The other of advantage of such a scheme is that it is generally quire fast. However, the scheme tends unwieldy when used to authenticate multiple messages because additional data needs to be generates to both sign and verify each new message. By contrast, with conventional signature schemes like RSA, the same key pair can be used to authenticate multiple documents, which will face the threat of replay attacks.

In this paper, we propose a new scheme to generalize the Lamport one-time signature. We group the message by power of 2. Then, each group of the message is encoded and signs individually by selecting the corresponding private keys from the private key box to create the signature. Thus, the proposed scheme saves on the storage space of the public keys and the size of the signatures. Moreover, we propose an efficient solution for signing a long message to make the proposed scheme more operative in practical.

2. Lamport's one-time signature

In this section, we briefly review the Lamport one-time signature, which includes three algorithms: key generation, signature and verification. Suppose that $f: Y \to Z$ is a one-way hash function.

2.1. Key generation

1. Select $2k$ elements $y_{i,j} \in Y$ randomly with $1 \leqslant i \leqslant k$ and $j = 1, 0$ where k is the length of message based on 2.
2. Compute $z_{i,j} = f(y_{i,j})$ for all i,j.

3. The key K consists of the $2k$ y's and $2k$ z's. The private key SK box and the public key PK box are shown as follows:

$$SK = \begin{pmatrix} y_{1,0} & y_{2,0} & \cdots & y_{k,0} \\ y_{1,1} & y_{2,1} & \cdots & y_{k,1} \end{pmatrix}, \quad PK = \begin{pmatrix} z_{1,0} & z_{2,0} & \cdots & z_{k,0} \\ z_{1,1} & z_{2,1} & \cdots & z_{k,1} \end{pmatrix}.$$

2.2. Signature

To sign a k-bit message $m = m_1 \ldots m_k$, we should do the following steps:

1. The corresponding entries of the message $m_1, \ldots, m_k$ are $y_{1,m_1}, \ldots, y_{k,m_k}$.
2. Define the signature
   $$\mathrm{sig}(m_1, \ldots, m_k) = (y_{1,m_1}, \ldots, y_{k,m_k}).$$
3. Select corresponding entries from the key box to create signature.

For example, we want to sign a message $m = 10 \ldots 1$. The signature is

$$sig(m_1, \ldots, m_k) = \begin{pmatrix} y_{1,0} & [y_{2,0}] & \cdots & y_{k,0} \\ [y_{1,1}] & y_{2,1} & \cdots & [y_{k,1}] \end{pmatrix} = \begin{pmatrix} y_{1,1} & y_{2,0} & \cdots & y_{k,1} \end{pmatrix}$$

on message $m_1, \ldots, m_k$.

2.3. Verification

To verify signature $(y_{1,1} y_{2,0} \ldots y_{k,1})$ on message $m_1, \ldots, m_k$, we check if

$$f(y_{i,m_i}) = z_{i,m_i} \text{ holds, where } 1 \leqslant i \leqslant k.$$

If the equation holds, we accept the signature otherwise reject it.

A message to be signed is a binary k-tuple. Each bit selects the corresponding value in the SK box as signed value. If the ith message bit is m_i the signature is y_{i,m_i} in the SK box. To verify the signature, we just check whether the hash value of each element is the corresponding value in the PK box. The Lamport one-time signature scheme faces the large length of signatures that is the half size of SK box. It is not implemented in practice.

3. The proposed scheme

We propose a new scheme to improve the size of Lamport one-time signature. Furthermore, the Lamport one-time signature scheme is just a special case

of the proposed scheme. The new scheme including key generation, signature and verification is described as follows.

3.1. Key generation

We should do the following steps:

1. Select a number e and set $v = 2^{e+1}$.
2. Based on v, encode message $m = (m_1, \ldots, m_l)_v$, where l is length of message after encoding.
3. For each column i, randomly select $e + 1$ elements $\in Y$ with suffix by power of 2 as $y_{i,2^0}, y_{i,2_1}, \ldots, y_{i,2^e}$, where $1 \leqslant i \leqslant l$ and Y is mentioned in Section 2.
4. Compute the corresponding public key box by using hash function.

Thus private key box SK and the public key PK box are shown as follows:

$$SK = \begin{pmatrix} y_{1,2^0} & y_{2,2^0} & \cdots & y_{l,2^0} \\ y_{1,2^1} & y_{2,2^1} & \cdots & y_{l,2^1} \\ \vdots & \vdots & \ddots & \vdots \\ y_{1,2^e} & y_{2,2^e} & \cdots & y_{l,2^e} \end{pmatrix}, \quad PK = \begin{pmatrix} z_{1,2^0} & z_{2,2^0} & \cdots & z_{l,2^0} \\ z_{1,2^1} & z_{2,2^1} & \cdots & z_{l,2^1} \\ \vdots & \vdots & \ddots & \vdots \\ z_{1,2^e} & z_{2,2^e} & \cdots & z_{l,2^e} \end{pmatrix}.$$

3.2. Signature

To sign the message m, we should do the following steps:

1. Encode the message based on v as $m = (m_1, \ldots, m_l)_v$.
2. Decode each digit m_i based on 2 as $m_i = (u_1, u_2, \ldots, u_e)_2$.
3. According to $m_i = (u_1, u_2, \ldots, u_e)_2$, select corresponding entries of Y at column i. For example, if $u_j = 1$ select $y_{i,2^j}$, else discard it, where $1 \leqslant j < e$.

Thus, the signature of message m is the selected items in the private key box $(a_1, a_2, \ldots, a_n)$.

To sign on message m for example, we select $e = 3$ and set $v = 2^{(3+1)} = 16$ and encode $m = (3A \ldots 1)_{16}$; then we decode $3 = (0011)_2$, $A = (1010)_2$ and $1 = (0001)_2$, etc. and select the corresponding entries in the private key box $(a_1, a_2, \ldots, a_n) = (y_{1,2^0}, y_{1,2^1}, y_{2,2^1}, {}_{2,2^3}, \ldots y_{1,2^0})$ as a signature

$$\mathrm{sig}(3A\ldots1) = \begin{pmatrix} [y_{1,2^0}] & y_{2,2^0} & \cdots & [y_{l,2^0}] \\ [y_{1,2^1}] & [y_{2,2^1}] & \cdots & y_{l,2^1} \\ y_{1,2^2} & y_{2,2^2} & \cdots & y_{l,2^2} \\ y_{1,2^3} & [y_{2,2^3}] & \cdots & y_{l,2^3} \end{pmatrix} = (y_{1,2^0}, y_{1,2^1}, y_{2,2^1}, y_{2,2^3}, \ldots, y_{l,2^0}).$$

3.3. Verification

To verify the signature $(a_1, a_2, \ldots, a_n)$ on message m. We hash each elements of the signature $(a_1, a_2, \ldots, a_n)$ and check whether equal to corresponding entries in public key PK box.

For example, we check the signature $(y_{1,2^0}, y_{1,2_1}, y_{2,2^1}, y_{2,2^3}, \ldots, y_{1,2^0})$ on message $m = (3A\ldots1)_{16}$; select the corresponding entries in the public key box and check that the pre-image of the selected entries are the signature as the follows:

$$f(y_{1,2^0}, y_{1,2^1}, y_{2,2^1}, y_{2,2^3}, \ldots, y_{l,2^0}) = \begin{pmatrix} [z_{1,2^0}] & z_{2,2^0} & \cdots & [z_{l,2^0}] \\ [z_{1,2^1}] & [z_{2,2^1}] & \cdots & z_{l,2^1} \\ z_{1,2^2} & z_{2,2^2} & \cdots & z_{l,2^2} \\ z_{1,2^3} & [z_{2,2^3}] & \cdots & z_{l,2^3} \end{pmatrix}.$$

In the proposed scheme, a message to be signed is based on the power of 2. The message is divided into l digits. Each digit of the message is signed individually. The signature is the corresponding entries of private key box with 1's binary in each digit encoded by based of 2. The verification is checking whether each items of signature is the pre-image of the corresponding public key entries.

4. Security analysis and performance

The adversary attempts to forge the proposed signatures. We show the adversary how to invert the proposed scheme. Assume the adversary forge the signature with probability δ. From the rows of the signature, the terms selected from the private key box are about $\frac{e}{2}$ so that to break the signature of row is with probability $\frac{2}{e}$. Because the signature has l columns in the signature box, the adversary forges the signature with $\frac{2}{e} * \frac{1}{l}$. We know to break the hash function f with a negligible probability ε such that we have $\frac{2}{e} * \frac{1}{l} * \varepsilon$ and therefore $\delta \cong \frac{2\varepsilon}{el}$. To break the proposed scheme is with probability around $\frac{2\varepsilon}{el}$.

Lamport's one-time signature scheme is not efficient on the length of signatures and public keys. The proposed scheme, moreover, saves the space storage of the signatures and the public keys. We consider that the length of message is

Table 1
Comparison of the proposed scheme and Lamport one-time signature scheme

	# Of public key items	Avg. # of signature terms	Avg. # of verification times
Lamport's signature	320	160	160
Proposed scheme based on 32	160	80	80
Improvement (based on 32)		50%	

160 bits and the proposed scheme based on 32. We compare the proposed scheme and the Lamport's one-time signature scheme in the Table 1.

We encode the message based on 32 and get $l = 160/5 = 32$. The number of items in each column is $e + 1 = 5$, since $e = 4$ deserves five items and the special item when the message digit is 0. The public key has $32*5 = 160$ items. The signature items are average of one half of the public key items. Therefore, we save about (320–160)/320 = 50% storage in storage space compared to the Lamport one-time scheme.

Although, the proposed scheme is better than the Lamport one-time signature scheme, it is still not efficient to sign a very long message. We may improve the problem by hashing the message before signing, i.e., to sign message m, first we compute the hash value $\tilde{m} = f(m)$ by using hash function, then we sign the message $\tilde{m}$ by the proposed scheme. The result leads that the signature is not increasing with the length of the message to be signed. We consider using the SHA-2 hash function in the proposed scheme such that the length of the message is 160 bits.

5. Conclusion

We have proposed the generalized Lamport one-time signature scheme which saves storage space. The proposed scheme is used to sign a long message by hashing the message before signing to make the proposed scheme more efficient. We expect that our scheme can be used to build more operative one-time signature schemes.

References

[1] M. Bellare, S. Micali, How to sign given any trapdoor function, Journal of Cryptology—Crypto'92, LNCS 740 (1993) 1–14.
[2] L. Lamport, Constructing digital signatures from a one-way function, Technical Report CSL-98, SRI International, 1979.
[3] National Institute of Standards and Technology (NIST), Secure hash standard, Federal Information Processing Standards Publication FIPS PUB 180-2, August 2001.

[4] R.C. Merkle, A certified digital signature based on a conventional function, in: Advances in Cryptology—Crypto'87, LNCS, 293, 1987, pp. 369–378.
[5] M.O. Rabin, Digitalized Signatures, Foundations of Secure Communication, Academic Press, New York, 1979, pp. 155–168.
[6] R. Rivest, The MD5 message-digest algorithm, Internet Request for Comment RFC 1321, Internet Engineering Task Force, April 1992.
[7] H. Wang, J. Pieprzyk, Efficient one-time proxy signatures, in: Asiacrypto 2003, LNCS 2894, 2003, pp. 507–522.
[8] T.C. Wu, H.S. Sung, An improved one-time digital signature scheme based on one-way function, Journal of Information Science and Engineering 12 (3) (1996) 387–395.
[9] K. Zhong, Efficient protocols for signing routing messages, in: Proceedings of the NDSS, 1998.