A Batch Verifying and Detecting Multiple RSA Digital Signatures

S. Wesley Changchien Min-Shiang Hwang

Department of Information ManagementChaoyang University of Technology
168, Gifeng E. Rd., Wufeng,
Taichung County, Taiwan 413, R.O.C.
Email: mshwang@cyut.edu.tw
http://www.cyut.edu.tw/∼mshwang/

March 4, 2002

Abstract: In this article, we propose an extension of Harn's proposal for an efficient, batch verification multiple RSA digital signature system. Our scheme can verify RSA signatures with the ability to detect forged signatures efficiently.

AMS Subject Classification:
Key Words: Cryptography, digital signatures, RSA.

1 Introduction

To reduce the signature verification time, Harn proposed two batch verifying multiple RSA and DSA-type digital signatures [3, 4]. However, his schemes are insecure [5, 6]. Recently, Harn proposed another efficient method to batch verify RSA signature's signed using the same private key [4]. Assume that the RSA signature scheme [1, 2, 7] has selected the following parameters.

- $n = pq$, where p and q are two large secret primes.
- e is the public key of a user, say Alice.
- d is a private key of the user (Alice), where $ed \bmod \phi(n) = 1$. Here, $\phi(\cdot)$ is a Euler totient function.

- $h(\cdot)$ is a one-way hash function.

The RSA signature of a message M_i is $S_i = h(M_i)^d \bmod n$. The signature verification is performed by checking whether $h(M_i)$ is equal to $S_i^e \bmod n$. Harn proposed an application in which a user, Alice, generates t signatures $(M_i, S_i), i = 1, 2, \cdots, t$. Alice sends these t signatures to the other user, Bob. In a straightforward scheme, Bob needs to verify all signatures one by one. Therefore, Bob needs to compute $h(M_i) = S_i^e \bmod n$ for $i = 1, 2, \cdots, t$. The computation requires t exponentiations. However, Bob only needs to compute $(\prod_{i=1}^t h(M_i) = (\prod_{i=1}^t S_i)^e \bmod n)$ in Harn's scheme. The computation requires only one exponentiation and $(t-1)$ multiplications. Since exponential computation is more time consuming than multiplicative computation, the performance of Harn's scheme is better than that of the straightforward scheme.

Harn demonstrated that if the multiplicative signature satisfies $(\prod_{i=1}^t S_i)^e = \prod_{i=1}^t h(M_i) \bmod n$, then Harn says that the multiplicative signature is a valid signature for messages M_1, M_2, $\cdots$, M_t.

2 An Extension Scheme

In Harn's scheme, Bob computes and checks whether $(\prod_{i=1}^t S_i)^e \bmod n$ is equal to $\prod_{i=1}^t h(M_i)$. If both are equal, the RSA signatures of M_1, M_2, $\cdots$, M_t are S_1, S_2, $\cdots$, S_t, respectively. Otherwise, there exists at least one signature for M_i that is not S_i for some $i = 1, 2, \cdots, t$. In this case, the verification of all individual signatures must be performed by checking whether $h(M_i)$ is equal to $S_i^e \bmod n$. It is inefficient to compute all of the individual signatures.

In this article, we extend Harn's scheme to reduce the computations when the batch verification fails, that is $(\prod_{i=1}^t S_i)^e \bmod n \neq \prod_{i=1}^t h(M_i)$. We redefine $h(\cdot)$ as a one-way hash function such that $h(\cdot)$ is a prime and $\prod_{i=1}^t h(M_i) \leq n$. We can let the length of $h(\cdot)$ is $\lfloor \frac{|n|}{t} \rfloor$ bits. Here, $\lfloor \cdot \rfloor$ denotes floor function; and $|n|$ denotes length of n. Other parameters and algorithms are the same as that in Harn's scheme. In case the batch verification fails, the forged signatures can be detected using the following steps.

1. Bob computes $L = (\prod_{i=1}^t S_i)^e \bmod n$.
2. Bob computes and verifies whether $L \bmod h(M_i) = 0$, for $i = 1, 2, \cdots, t$. If true then the signature of M_i is S_i, otherwise the S_i is a forged signature of M_i.

If RSA signatures of M_1, M_2, $\cdots$, M_t are S_1, S_2, $\cdots$, S_t, respectively,

$$\begin{aligned} & L \bmod h(M_i) \\ = \; & ((\prod_{i=1}^t S_i)^e \bmod n) \bmod h(M_i) \\ = \; & (\prod_{i=1}^t h(M_i)) \bmod h(M_i) \\ = \; & 0. \end{aligned}$$

If RSA signatures of M_i' is S_i and $M_i' \neq M_i$ (that is, the signature of M_i is not S_i),

$$\begin{aligned}
& L \bmod h(M_i) \\
= \ & ((\prod_{i=1}^{t} S_i)^e \bmod n) \bmod h(M_i) \\
= \ & (h(M_1)h(M_2)\cdots h(M_{i-1})h(M_i')h(M_{i+1})\cdots h(M_t)) \bmod h(M_i) \\
\neq \ & 0.
\end{aligned}$$

Since $h(\cdot)$ is a prime, it is not exist M_j such that $h(M_j) \bmod h(M_i) = 0$, for all j and $j \neq i$.

Instead of computing t exponential computations, our scheme only requires the computation of one exponentiation and t modulus.

3 Conclusion

We have proposed an efficient method for to batch verifying RSA signatures with the capability to detect and identify forged signatures.

Acknowledgements

Part of this research was supported by the National Science Council, Taiwan, R.O.C., under contract no. NSC90-2213-E-324-005.

References

[1] Chin-Chen Chang and Min-Shiang Hwang. Parallel computation of the generating keys for RSA cryptosystems. *IEE Electronics Letters*, 32(15):1365–1366, 1996.

[2] T. ElGamal. A public-key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, IT-31(4):469–472, July 1985.

[3] L. Harn. Batch verifying multiple DSA-type digital signatures. *Electronics Letters*, 34(9):870–871, 1998.

[4] L. Harn. Batch verifying multiple RSA digital signatures. *Electronics Letters*, 34(12):1219–1220, 1998.

[5] M. S. Hwang, I. C. Lin, and K. F. Hwang. Cryptanalysis of the batch verifying multiple RSA digital signatures. *Informatica*, 11(1):15–19, 2000.

[6] Min-Shiang Hwang, Cheng-Chi Lee, and Eric Jui-Lin Lu. Cryptanalysis of the batch verifying multiple DSA-type digital signatures. *Pakistan Journal of Applied Sciences*, 1(3):287–288, 2001.

[7] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public key cryptosystems. *Communications of the ACM*, 21(2):120–126, Feb. 1978.