Cryptanalysis and Improvement on Batch Verifying Multiple RSA Digital Signatures*

Feng Bao[§] Cheng-Chi Lee[‡] Min-Shiang Hwang[†]

Department of Management Information System[†]
National Chung Hsing University
250 Kuo Kuang Road,
402 Taichung, Taiwan, R.O.C.
Email: mshwang@nchu.edu.tw

Department of Computer Science[‡]
National Chung Hsing University
250 Kuo Kuang Road, 402 Taichung, Taiwan, R.O.C.
and Department of Computer & Communication Engineering
Taichung Healthcare and Management University
No. 500, Lioufeng Raod, Wufeng Shiang, Taichung, Taiwan, R.O.C.

Laboratories for Information Technology[§]
21 Heng Mui Keng Terrace,
Singapore 119613
Email: baofeng@krdl.org.sg

February 5, 2005

Cryptanalysis and Improvement on Batch Verifying Multiple RSA Digital Signatures

Abstract

Recently, Hwang et al. proposed a simple batch verifying multiple RSA digital signatures. Their scheme is efficient to reduce computation of verifying multiple RSA signatures. In this article, we propose a cryptanalysis and improvement of their scheme.

Keywords: Cryptography, digital signature, RSA, security, batch verifying.

1 Introduction

Public key cryptosystem was an epoch-creating approach of research in the last few decades, and it is still drawing hundreds and thousands of researchers' attention now. In such a cryptosystem, each valid user has a public key and a private key. One can either encrypt the content of the message with the receiver's public key or sign it with one's own private key to prove the validity of the access to the message. In other words, the receiver can either decrypt the ciphertext with his/her own private key or verify the sender's identity (signature) with the sender's public key. No user can forge the legitimate signature of another valid user when the private key of the user is not to be known. The most popular public key cryptosystems are RSA cryptosystem [?, ?, ?] and ElGamal cryptosystem [?, ?].

We briefly review the RSA cryptosystem [?, ?, ?] as follows. The cryptosystem has two large primes p and q, and the modulus $n = p \times q$. Give $e \times d \bmod (p-1)(q-1) = 1$, where e is the public key and d is the private key.

*This research was partially supported by the National Science Council, Taiwan, R.O.C., under contract no.: NSC92-2213-E-005-027.

Assume that a signer wants to send message m and its signature s to a receiver. The signer uses the RSA signature algorithm to sign the message m. The signing procedure is $s = h(m)^d \bmod n$, where d is the signer's private key and $h(\cdot)$ is a public one-way hash function. Then, the signer sends the pair (m, s) to her/his receiver. Whenever the receiver receives (m, s), she/he can verify the correctness of the signature on the message m by checking $h(m) = s^e \bmod n$, where e is the signer's public key. Now, assume that a signer sends t pairs (m, s)'s to a receiver, and then the receiver will have to verify the correctness of the multiple digital signatures by using t verifications. This is inefficient.

In 1994, Naccache et al. [?] proposed an efficient scheme to batch verify multiple DSA digital signatures. The multiple DSA digital signatures are batch verified by the receiver, and only one times verification is required (instead of t times verifications). However, this scheme is insecure [?]. In 1995, Harn proposed a DSA-type secure interactive batch verification protocol [?]. In 1998, Harn proposed two efficient non-interactive batch verification protocols: DSA-type and RSA-type multiple digital signatures [?, ?]. However, the batch verifying multiple RSA-type digital signatures [?] is insecure [?].

Recently, Hwang et al. have proposed two simple batch-verifying multiple digital signatures: BV-DSA and BV-RSA schemes [?]. The BV-RSA scheme can remedy the weakness pointed out in [?]. However, the BV-RSA scheme has the same security flaw as [?] that a dishonest signer can forge individual digital signatures and make false batch verifications valid. This weakness can lead the signer to deny his/her signed messages, which violates the non-repudiation property of digital signature. We will show that the probability of successful cheating is as high as 50%. Next, in this article, we shall propose a cryptanalysis and improvement on the BV-RSA scheme of Hwang et al.

2 Cryptanalysis of Hwang et al.'s Scheme

Recently, Hwang et al. have proposed two simple batch-verifying multiple digital signatures [?]: one is the BV-DSA scheme, and the other is the BV-RSA scheme. Here, we will introduce BV-RSA scheme as follows. Assume that a signer, Alice, wants to send messages $m_1, m_2, \cdots, m_t$ and their signatures $s_1, s_2, \cdots, s_t$, respectively, to a receiver, Bob. The scheme uses the same system parameters (e, d, n, p, q) as the RSA scheme does.

(1) Alice generates multiple digital signatures $(s_1, s_2, \cdots, s_t)$ and sends Bob

$$s_i \quad = \quad h(m_i)^d \bmod n, \text{ and } m_i, i = 1, 2, \cdots, t.$$

(2) After receiving these multiple signatures from Alice, Bob verifies the batch of these multiple signatures using the following equation:

$$(\prod_{i=1}^{t} s_i^{v_i})^e = \prod_{i=1}^{t} h(m_i)^{v_i} \bmod n, \tag{1}$$

where v_i, $i = 1, 2, \cdots, t$, are small random numbers which are randomly chosen by Bob.

In this scheme, Bob can verify these multiple digital signatures with Alice's public key, and only one times verification is required instead of t verifications. Hence, the BV-RSA scheme is simple and efficient to verify multiple RSA digital signatures. However, there is a weakness in this scheme. A dishonest signer, Alice, can forge individual digital signatures and make a false batching verification valid.

The cryptanalysis is as follows. A dishonest signer (Alice) can choose w such that $w^2 = 1 \bmod n$ since she knows p and q. And then she sends messages and the forged digital signatures (m_i, s_i'), $i = 1, 2, \cdots, t$, to a verifier (Bob), where

$$s_i' = s_i \times w \bmod n, i = 1, 2, \cdots, t.$$

The probability of the sum of v_i being even is about fifty percent, and $w^2 = 1 \bmod n$. Bob is therefore convinced that these messages where signed by Alice by verifying Equation (1). However, when a dispute occurs, Alice can deny her signing messages because $h(m_i) \neq (s'_i)^e \bmod n$. This violates the non-repudiation property of digital signatures. The odds of successful cheating is 1/2 since all v_i's are randomly chosen, and the subset can be randomly chosen by the verifier.

3 Our Scheme

In the above section, we have introduced the weaknesses of the BV-RSA scheme that it cannot achieve non-repudiation. A dishonest signer can forge individual digital signatures and make a false batching verification valid. For batch verification, we need to guarantee: As long as some signatures can pass batch verification, they should be valid ones; i.e., they should pass individual verification. Our scheme can achieve non-repudiation because: without a private key, no one can forge s'_i (actually s'_i is also a valid signature in our scheme). If someone can generate s'_i, he/she must own a private key. Therefore, our scheme can achieve non-repudiation.

To remedy the weaknesses of BV-RSA, we first modify the signature verification of RSA a bit. We replace the verification formula $h(m_i) = (s_i)^e \bmod n$ with $h(m_i)^2 = (s_i)^{2e} \bmod n$. Our scheme uses the same system parameters (e, d, n, p, q) as the RSA scheme. Assume that the signer is Alice and the verifier is Bob.

(1) Alice generates multiple digital signatures $(s_1, s_2, \cdots, s_t)$ and sends Bob

$$s_i = h(m_i)^d \bmod n, \text{ and } m_i, i = 1, 2, \cdots, t.$$

(2) After receiving these multiple signatures from Alice, Bob verifies the

batch of these multiple signatures using the equation:

$$(\prod_{i=1}^{t} s_i^{v_i})^{2e} = \prod_{i=1}^{t} h(m_i)^{2v_i} \bmod n, \tag{2}$$

where v_i, $i = 1, 2, \cdots, t$, are small random numbers which are randomly chosen by Bob.

Next, we analyze the security of our improved schemes as follows. Based on Harn's scheme [**?**] and Hwang et al.'s scheme [**?**], the security of our improved scheme is the same as that of those schemes. Furthermore, the above attack does not work on our scheme. As long as the signatures pass our batch verification formula, they must be correct signatures. Hence, when a dispute occurs, Alice cannot deny her having signed messages because $h(m_i)^2 = (s_i')^{2e} \bmod n$. This satisfies the non-repudiation property of digital signatures.

In addition, another malicious attack should be taken into account, too. The above improvement is based on the assumption that the signer will not generate "bad" public/private keys. In other words, we assume that $n = p \times q$ and $p = 2p' + 1$ and $q = 2q' + 1$; i.e., p and q are safe primes (here p' and q' are primes too). However, if the signer deliberately chooses special p and q, he/she can still fool the verifier. What he/she can do is to choose p such that $p - 1$ has a small prime factor other than 2, say, $p = 6p' + 1$ where p' is a prime. In this case, $(p - 1)(q - 1)$ has factor 3. Then the signer can choose w such that $w^3 = 1 \bmod n$. Again he/she multiplies w to a subset of signatures.

For example, assume $s_1' = ws_1$, $s_3' = ws_3$, $s_6' = ws_6$, and $s_7' = ws_7$. In this case, s_1', s_3', s_6', and s_7' are all wrong signatures because $h(m_i)^2 \neq (s_i')^{2e} \bmod n$. But they can pass our batch verification formula using Equation (2) if $v_1 + v_3 + v_6 + v_7$ is a multiple of 3. Since all v_is are randomly chosen and the subset can be randomly chosen by the verifier, the odds of successful cheating is 1/3. Fortunately, to prevent such cheating, we can ask the signer to prove $p = 2p' + 1$ and $q = 2q' + 1$. There are papers on zero-knowledge proof of $p = 2p' + 1$ and $q = 2q' + 1$ without disclosing p and q [**?**, **?**].

4 Discussion and Conclusion

In general, the probability of successful cheating is not only as high as 1/2, but also $1/a$, where $a \geq 2$. The reason is as follows. A dishonest signer can choose w such that $w^a = 1 \bmod n$ since she/he knows p and q. Then the signer can forge individual digital signature and make a false batch verification valid since the probability of the sum of v_i can be divided by a. To enhance the security of the scheme, we can modify our batch verifying formula as $(\prod_{i=1}^{t} s_i^{v_i})^{be} = \prod_{i=1}^{t} h(m_i)^{bv_i} \bmod n$, where $b = \prod_{j=2}^{a} j$.

Batch Verifying multiple digital signatures are efficient for verifying multiple digital signatures. The verifier can verify multiple digital signatures with the signer's public key, and only one times verification is needed instead of t verifications. In this article, we have shown our cryptanalysis on Hwang et al.'s scheme and proposed an improvement. Our scheme can enhance the security of the batch verifying multiple RSA digital signatures.