Improvement on a Digital Signature Scheme without using One-way Hash and Message Redundancy

Jie Liu and Jianhua Li
Department of Electronic Engineering, Shanghai Jiao Tong University
ljiesh@gmail.com

Abstract

Digital signature schemes based on public-key cryptosystems generally permit existential forgery, except the schemes are equipped with some message formatting mechanisms, such as using hash functions or padding redundancies. In 2004, Chang et al. proposed a new digital signature scheme, and claimed the scheme without using any hash function or padding any redundancy can resist forgery attacks. However, many attacks on Chang et al.'s scheme were presented. Kang et al. also gave an effective improvement to resist these forgery attacks. In this letter, we gave a further improvement to shorten the signed signature. Our improvement keeps the security of Kang et al.'s scheme and makes it more efficient in computation and communication.

Keywords: *digital signature; message recovery; forgery attack; one-way hash; message redundancy*

1. Introduction

Digital signature schemes based on public-key cryptosystems generally permit existential forgery. A usual method to prevent existential forgery is using message formatting mechanisms, such as using hash functions and padding redundancies, which permit a verifier to verify non-random distribution of a message. The ElGamal signature family is a typical example. The message signed in the original ElGamal signature scheme does not contain any recognizable redundancy and the scheme is subject to existential forgery [1]. However, the modified ElGamal signature scheme, into which a hash function is introduced, was proved secure by Pointcheval and Stern [5]. Schnorr signature scheme [2] and Digital Signature Standard (DSS) [3] are another two influential variations in ElGamal-family signatures. Both of them utilize hash functions and can resist forgery attacks.

Commonly in a digital signature scheme, the signed signature is appended to the original message and sent to the receiver together. Then the receiver can verify the validity of the signed message using the signer's public key. Another kind of digital signature scheme is with message recovery, where the original message is included and hidden in the signed signature [4]. Upon receiving the signature, the receiver can recover the original message from the signature and complete the verification process. A digital signature with message recovery is more useful for some applications in which the message to be signed is small, such as a time, numbers, and so on.

In 2004, Chang *et al.* [6] proposed a digital signature scheme with message recovery and claimed that their scheme can resist forgery attacks without using any hash function or padding any redundancy. However, many papers proved that Chang *et al.*'s scheme was insecure. Zhang [8], Fu *et al.* [7] and Chien [9] gave their forgery attacks on Chang *et al.*'s scheme recently. Kang *et al.* gave an effective improvement to resist these forgery attacks [10]. In this letter, we proposed two more efficient schemes. Our schemes are as secure as Kang *et al.*'s scheme. However, the signature generated in our scheme is much shorter than Kang *et al.*' scheme. The shortened signature means fewer operations in signature generation and verification and less transmission in signature sending. That is to say, we make the signature schemes more efficient in computation and communication.

2. Review of Chang *et al.*'s Scheme

The initialization for Chang *et al.*'s digital signature scheme is described as follows. Let p be a large prime, and $g \in \mathbb{F}_p^*$ is a random multiplicative generator element. x is the private key of the signer U ,where $x < p-1, \gcd(x, p-1) = 1$. Y is the corresponding public key such that $Y \equiv g^x \bmod p$. To generate a signature for message $m \in \mathbb{Z}_p$, U executes the following steps.

A. Signature-Generation Phase

1. U computes $s \equiv Y^m \bmod p$.
2. U randomly chooses $k \in [1, p-1]$ and computes
$$r \equiv m \times s \times g^{-k} \bmod p$$
3. U derives the value t from
$$s + t \equiv x^{-1} \times (k - r) \bmod (p-1)$$
4. U sends the signature (r, s, t) of m to the verifier.

B. Verification Phase Upon receiving the signature (r, s, t) of m , the verifier V performs the following steps to validate the signature.

1. V computes
$$m^* \equiv Y^{s+t} \times r \times g^r \times s^{-1} \bmod p$$
2. V checks whether $s \equiv Y^{m^*} \bmod p$ holds. If so, he accepts the signature; otherwise, he rejects it.

It's easy to verify that Chang *et al.*'s digital signature scheme works correctly.

3. Forgery Attacks on Chang *et al.*'s Scheme

Assume Eve is an attacker who wants to forge signature (r', s', t') for message m'. (r, s, t) is a legitimate signature for message m, which may be utilized in Eve' forge process.

3.1. Fu *et al.*'s Forgery Attack 1

Eve randomly chooses $r' \in \mathbb{Z}_p^*$ and computes:
$$m' \equiv r' \cdot g^{r'} \bmod p$$
$$s' \equiv Y^{m'} \bmod p$$
$$t' \equiv (m' - s') \bmod (p-1)$$
(r', s', t') can pass the verification because:
$$\begin{aligned} m^* &\equiv Y^{s'+t'} r' g^{r'} (s')^{-1} \\ &\equiv Y^{m'} r' g^{r'} Y^{-m'} \\ &\equiv m' \bmod p \end{aligned}$$

3.2. Fu *et al.*'s Forgery Attack 2

Eve randomly chooses $R \in \mathbb{Z}_p^*$ and computes:
$$r' \equiv r \cdot R \bmod p$$
$$m' \equiv mRsg^{r(R-1)} \bmod p$$
$$s' \equiv Y^{m'} \bmod p$$
$$t' \equiv (s + t + m') - s' \bmod (p-1)$$
(r', s', t') can pass the verification because:
$$\begin{aligned} m^* &\equiv Y^{s'+t'} r' g^{r'} (s')^{-1} \\ &\equiv Y^{s+t+m'} rRg^{rR} Y^{-m'} \\ &\equiv (Y^{s+t} rg^r s^{-1}) Rg^{rR-r} s \\ &\equiv m' \bmod p \end{aligned}$$

3.3. Zhang's Forgery Attack 1

Eve randomly chooses $\alpha \in \mathbb{Z}_{p-1}^*$ and computes:
$$m' \equiv m \cdot Y^{\alpha} \bmod p$$
$$s' \equiv Y^{m'} \bmod p$$
$$r' \equiv r \bmod p$$
$$t' \equiv s + t - m + \alpha - s' + m' \bmod (p-1)$$
(r', s', t') can pass the verification because:
$$\begin{aligned} m^* &\equiv Y^{s'+t'} r' g^{r'} (s')^{-1} \\ &\equiv Y^{s+t-m+\alpha+m'} rg^r Y^{-m'} \\ &\equiv (Y^{s+t} rg^r s^{-1}) Y^{\alpha} \\ &\equiv m' \bmod p \end{aligned}$$

3.4. Zhang's Forgery Attack 2

Eve randomly chooses $\alpha \in \mathbb{Z}_p^*$ and computes:
$$r' \equiv \alpha \cdot r \bmod p$$
$$\beta + r \equiv r' \bmod (p-1), \beta \in \mathbb{Z}_{p-1}^*$$
$$m' \equiv m \cdot \alpha \cdot g^{\beta} \bmod p$$
$$s' \equiv Y^{m'} \bmod p$$
$$t' \equiv s + t - m - s' + m' \bmod (p-1)$$
(r', s', t') can pass the verification because:
$$\begin{aligned} m^* &\equiv Y^{s'+t'} r' g^{r'} (s')^{-1} \\ &\equiv Y^{s+t-m+m'} \alpha rg^{r+\beta} Y^{-m'} \\ &\equiv (Y^{s+t} rg^r s^{-1}) \cdot \alpha \cdot g^{\beta} \\ &\equiv m' \bmod p \end{aligned}$$

3.5. Chien's Attack

Eve randomly chooses $k' \in [1, p-1]$ and computes:
$$r' \equiv Y^{k'} \bmod p$$
$$m' \equiv g^{r'} \bmod p$$
$$s' \equiv Y^{m'} \bmod p$$
$$t' \equiv -s' - k' + m' \bmod (p-1)$$
(r', s', t') can pass the verification because:
$$\begin{aligned} m^* &\equiv Y^{s'+t'} r' g^{r'} (s')^{-1} \\ &\equiv Y^{-k'+m'} \cdot Y^{k'} \cdot g^{r'} \cdot Y^{-m'} \\ &\equiv m' \bmod p \end{aligned}$$

4. Kang *et al.*' Improvement

Kang *et al.* gave a detailed cryptanalysis on above forgery attacks and summarized a common attack named *parameter reduction attack*, which revealed the fatal flaw of Chang *et al.*'s signature scheme.

Let $t' = s + t - m$, the verification phase of Chang *et al.*'s scheme can be transformed as:
$$\begin{cases} m^* \equiv Y^{t'} \times r \times g^r \bmod p \\ s = Y^{m^*} \bmod p \end{cases}$$
The parameters in the equation to recover the original message are reduced from three to two. Therefore, the attacker can choose arbitrary r, t' and compute m, then compute s. This kind of signature $(r, s, t' - s + m)$ for

message m always can pass the verification. Chang *et al.* hoped to resist forgery attacks utilizing an extra parameter $s \equiv Y^m \bmod p$ in their scheme other than using hash functions or padding redundancies. However, this parameter s does not work as what Chang *et al.*'s hoped. It's useless in the equation to recover the original message. That's the fatal flaw of Chang *et al.*'s signature scheme.

Then Kang *et al.* gave an effective solution to fix this flaw in [10]. To sign message $m \in \mathbb{Z}_p$, U executes the following steps.

A. Signature-Generation Phase

1. U computes $s \equiv Y^m \bmod p$.
2. U randomly chooses $k \in [1, p-1]$ and computes
$$r \equiv s + mg^{-k} \bmod p.$$
3. U derives the value t from
$$s + t \equiv x^{-1} \times (k - r) \bmod (p-1)$$
4. U sends the signature (r, s, t) of m to the verifier.

B. Verification Phase Upon receiving the signature (r, s, t) of m, V performs the following steps to validate the signature.

1. V computes
$$m^* \equiv (r - s) Y^{s+t} g^r \bmod p$$
2. V checks whether $s \equiv Y^{m^*} \bmod p$ holds. If so, he accepts the signature; otherwise, he rejects it.

Kang *et al.*'s scheme works correctly since:

$$\begin{aligned} m^* &\equiv (r - s) Y^{s+t} g^r \\ &\equiv mg^{-k} g^{k-r} g^r \\ &\equiv m \bmod p \end{aligned}$$

Let $t' = t + s$ and $r' = r - s$, the equation to recover the original message in Kang *et al.*' improved scheme can be transformed as:

$$m^* \equiv (r - s) Y^{t'} g^r \bmod p$$
$$m^* \equiv r' Y^{t'} g^{r'+s} \bmod p$$

The transformation is not able to reduce parameters as what Kang *et al.* did to Chang *et al.*'s scheme. That is to say, Kang *et al.*' improved scheme can resist the so-called *parameter reduction attack*, including those attacks depicted in part 3.

5. Our Improved Schemes

Kang *et al.*' improved scheme is secure to resist forgery attacks. However, we found that it is not efficient enough. To sigh a message m of length $|p|$, the length of the signature is about $3|p|$. The longer signature means more operations in signature generation and verification and more transmission in signature sending. While preserving the security properties, the shortened signature will make the signature scheme more efficient.

5.1. Improvement on Kang *et al.*'s Scheme

Setup two prime numbers p and q such that $q \mid p-1$, where the typical size for these parameters are: $|p| = 1024$ and $|q| = 160$. Setup an element $g \in \mathbb{Z}_p^*$ of order q. x is the private key of the signer U ,where $x < p-1$, $\gcd(x, p-1) = 1$. Y is the corresponding public key such that $Y \equiv g^x \bmod p$. To generate a signature for message $m \in \mathbb{Z}_p$, U executes the following steps.

A. Signature-Generation Phase

1. U computes $s \equiv Y^m \bmod q$.
2. U randomly chooses $k \in [1, q]$ and computes
$$r \equiv s + mg^{-k} \bmod p$$
3. U derives the value t from
$$s + t \equiv x^{-1} \times (k - r) \bmod q$$
4. U sends the signature (r, s, t) of m to the verifier.

B. Verification Phase Upon receiving the signature (r, s, t) of m, V performs the following steps to validate the signature.

1. V computes
$$m^* \equiv (r - s) Y^{s+t} g^r \bmod p$$
2. V checks whether $s \equiv Y^{m^*} \bmod q$ holds. If so, he accepts the signature; otherwise, he rejects it.

Obviously, the improved scheme works correctly too.

5.2. Another Improved Signature Scheme

The system initialization is identical as described in 5.1. To generate a signature for message $m \in \mathbb{Z}_p$, U executes the following steps.

A. Signature-Generation Phase

1. U computes $s \equiv Y^m \bmod q$.
2. U randomly chooses $k \in [1, q]$ and computes
$$r \equiv mg^{-k} \bmod p$$
3. U derives the value t from
$$t \equiv x^{-1} \times (k - rs) \bmod q$$
4. U sends the signature (r, s, t) of m to the verifier.

B. Verification Phase Upon receiving the signature (r, s, t) of m, V performs the following steps to validate the signature.

1. V computes
$$m^* \equiv Y^t r g^{rs} \bmod p$$
2. V checks whether $s \equiv Y^{m^*} \bmod q$ holds. If so, he accepts the signature; otherwise, he rejects it.

We now show that (r, s, t) is a valid signature of message m in our improved scheme because:

$$\begin{aligned} m^* &\equiv Y^t r g^{rs} \\ &\equiv g^{k-rs} m g^{-k} g^{rs} \\ &\equiv m \bmod p \end{aligned}$$

5.3. Analysis of Our Improved Scheme

First, only some module values are changed in our improvement on Kang *et al.*'s scheme. So our improvement does not change the security properties of Kang *et al.*'s scheme. Thus, our improved scheme still can resist *parameter reduction attack*.

Secondly, the signature scheme in 5.2 is secure against *parameter reduction attack* as Kang *et al.*'s scheme. We have already made a reduction in deriving the value t. We omit the useless s so the parameter t is not reducible any longer. When let $r' = rs$, the equation to recover the message can be transformed as:

$$m^* \equiv Y^{t'} \left(r' s^{-1} \right) g^{r'} \bmod p$$

This transformation is not able to reduce parameters either.

Thirdly, also our main contribution, the signature generated in our improved schemes is much shorter than Chang *et al.*'s scheme and Kang *et al.*'s scheme: $(|p|+2|q|)$ bits are required for transmitting our signature, in comparison with $3|p|$ bits for transmitting a Chang *et al.*'s signature or a Kang *et al.*'s signature. As we mentioned above, shortened signature also means fewer operations in signature generation and verification: $O\left(\log^2 p \cdot \log^3 q\right)$ in our schemes vs. $O\left(\log^5 p\right)$ in Chang *et al.*'s scheme or Kang *et al.*'s scheme.

6. Conclusion

In this paper, we presented a further cryptanalysis of Chang *et al.*'s digital signature scheme, which was claimed to resist forgery attacks without using any one-way hash function or padding any redundancy. We reviewed some forgery attacks and Kang's improvement. Then, we proposed two improved signature schemes, in which the length of the signed signature is much shorter. Our improvement makes the schemes more efficient in computation and communication. Whether we can let modulo p part in signature generation be conducted in an off-line manner is an interesting open question. Such a design arrangement will make the signature schemes more suitable for a small device to perform.

7. References

[1] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms", *IEEE Transactions on Information Theory*, vol. IT-31, no. 4, pp. 469-472, 1985.

[2] C. P. Schnorr, "Efficient signature generation for smart cards", *Journal of Cryptology*, vol. 4, no. 3, pp. 161-174, 1991.

[3] NIST, "Digital Signature Standard", 1994

[4] K. Nyberg and R. A. Rueppel, "Message Recovery for Signature Schemes Based on the Discrete Logarithm Problem", in *Proc. of Eurocrypt94*, LNCS 950, pp. 182-193, Springer-Verlag, 1995.

[5] D. Pointcheval and J. Stern, "Security proofs for signature schemes", in *Advances in Cryptology - Eurocrypt 1996*, LNCS 1070, pp. 387-398, Springer-Verlag, 1996.

[6] C. C. Chang and Y. F. Chang, "Signing a digital signature withouting using one-way hash functions and message redundancy schemes", *IEEE Communication Letters*, vol. 8, no. 8, pp. 485-487, 2004.

[7] X. T. Fu, C. X. Xu, and G. Z. Xiao, "Forgery Attacks on Chang et al.'s signature scheme with message recovery", *http://eprint.iacr.org/2004/236*, 2004.

[8] F. G. Zhang, "Cryptanalysis of Chang et al.'s Signature Scheme with Message Recovery", *IEEE Communication Letters*, vol. 9, no. 4, pp. 358-359, 2005.

[9] H. Y. Chien, "Forgery Attacks on Digital Signature Schemes without using One-way Hash and Message Redundancy", *IEEE Communication Letters*, vol. 10, no. 324-325, 2006.

[10] L. Kang and X. H. Tang, "Digital signature scheme without hash functions and message redundancy", *Journal on Communications (In Chinese)*, vol. 27, no. 5, pp. 18-20, 2006.