A Signcryption Scheme with Signature Directly Verifiable by Public Key

Feng Bao and Robert H. Deng

Institute of Systems Science
National University of Singapore
Kent Ridge, Singapore 119597
Email: {baofeng, deng}@iss.nus.sg

Abstract. Signcryption, first proposed by Zheng [4, 5], is a cryptographic primitive which combines both the functions of digital signature and public key encryption in a logical single step, and with a computational cost siginficantly lower than that needed by the traditional signature-then-encryption approach. In Zheng's scheme, the signature verification can be done either by the recipient directly (using his private key) or by engaging a zero-knowledge interative protocol with a third party, without disclosing recipient's private key. In this note, we modify Zheng's scheme so that the recipient's private key is no longer needed in signature verification. The computational cost of the modified scheme is higher than that of Zheng's scheme but lower than that of the signature-then-encryption approach.

1 Introduction

To guarantee unforgibility, integrity and confidentiality of communications, the traditional method is to digitally sign a message then followed by public key encryption. Signcryption, first proposed by Zheng [4], is a new cryptographic primitive which simultaneously fulfills both the functions of signature and encryption in a single logical step, and with a computational cost significantly lower than that required by the traditional signature-then-encryption approach. Two types of signcryption schemes were studied in [4], one based on the ElGamal type of public key cryptosystems and the other on RSA. In this note, we only consider signcryption schemes based on the ElGamal type of public key cryptosystems.

In the signature-then-encryption approach using DSA for signature and the ElGamal public key cryptosysem for encryption, a total of 6 exponential computations are required: one for signature generation, two for encryption, one for decryption, and two for signature verification. Zheng's signcryption scheme needs only a total of 3 exponential computations: one for signcrypting and two for unsigncrypting.

In the traditional signature-then-encryption approach, the message and the sender's signature are obtained after the decryption by using the recipient's private key. The signature is then verified using only the sender's public key. Hence, the validity of the signature can be checked by anyone who has knowledge of the sender's public key. In the signcryption scheme of [4], the unsigncryption (decryption and signature verification) needs the recipient's private key; therefore, only the recipient can verify the signature. As pointed out in [5], the constraint of using the recipient's private key in unsigncryption is acceptable for certain applications where the recipient need not pass the signature to others for verification; however, Zheng's singcryption schemes can not be used in applications where a signature need to be validated by a third party only using the public key as in usual signature scheme. To overcome this problem, some methods are given in [5] by introducing an independent judge. That is done by proving the equality of discrete logarithms. However, [2] points out that the confidentiality is lost in this case.

In this note, we modify Zheng's signcryption scheme such that verification of a signature no longer needs the recipient's private key. Hence, the modified scheme functions in exactly the same manner as that of the signature-then-encryption approach. However, the modified scheme is not as efficient as Zheng's original scheme. Nevertheless, the modified scheme is still more computational efficient than the signature-then-encryption approach.

2 Zheng's Scheme

Task: Alice has a message m to send to Bob. Alice signcrypts m so that the effect is similar to signature-then-encryption.

Public Parameters

- p: a large prime.
- q: a large prime factor of $p-1$.
- g: an element of $\mathbf{Z}_p^*$ of order q.
- $hash$: a one-way hash function.
- KH: a keyed one-way hash function.
- (E, D): the encryption and decryption algorithms of a symmetric key cipher.

Alice's keys

- x_a: Alice's private key, $x_a \in \mathbf{Z}_q^*$.
- y_a: Alice's public key, $y_a = g^{x_a} \bmod p$.

Bob's keys

- x_b: Bob's private key, $x_b \in \mathbf{Z}_q^*$.
- y_b: Bob's public key, $y_b = g^{x_b} \bmod p$.

Signcrypting:

Alice randomly chooses $x \in_R \mathbf{Z}_q^*$, then sets
$(k_1, k_2) = hash(y_b^x \bmod p)$
$c = E_{k_1}(m)$
$r = KH_{k_2}(m)$

$s = x/(r + x_a) \bmod q$

Alice sends (c, r, s) to Bob.

Unsigncrypting:

Bob computes
$(k_1, k_2) = hash((y_a g^r)^{s x_b} \bmod p)$
$m = D_{k_1}(c)$ to recover the plaintext message,
and then checks whether $KH_{k_2}(m) = r$ for signature verification.

In [4], two singcryption schemes were given, called SDSS1 and SDSS2. Here we only describe the case for SDSS1. The case for SDSS2 is similar. In Zheng's **unsigncrypting** process, it is straightforward to see that x_b is involved for signature verification.

3 The Modified Singcryption Scheme

Using the same set of notations as in the last section for public parameters, Alice's key and Bob's keys, the modified scheme is described as follows:

Signcrypting:

Alice randomly chooses $x \in_R \mathbf{Z}_q^*$, then sets
$k_1 = hash(y_b^x \bmod p)$,
$k_2 = hash(g^x \bmod p)$
$c = E_{k_1}(m)$
$r = KH_{k_2}(m)$
$s = x/(r + x_a) \bmod q$.

Alice sends (c, r, s) to Bob.

Unsigncrypting:

Bob computes
$t_1 = (y_a g^r)^s \bmod p$
$t_2 = t_1^{x_b} \bmod p$
$k_1 = hash(t_2)$
$k_2 = hash(t_1)$
$m = D_{k_1}(c)$ to obtain the plaintext message,
then checks whether $KH_{k_2}(m) = r$ for signature verification.

Later when necessary, Bob may forward (m, r, s) to others, who can be convinced that it came originally from Alice by verifying

$$k = hash((y_a g^r)^s \bmod p) \text{ and } r = KH_k(m)$$

Actually, we can use *hash* to replace the keyed hash function KH. In that case, the modified scheme can be written as follows:

Modified Scheme

Signcrypting:

Alice randomly chooses $x \in_R \mathbf{Z}_q^*$, then sets
$t_1 = g^x \bmod p$
$t_2 = y_b^x \bmod p$
$c = E_{hash(t_2)}(m)$
$r = hash(m, t_1)$
$s = x/(r + x_a) \bmod q$

Alice sends (c, r, s) to Bob.

Unsigncrypting:

Bob computes
$t_1 = (y_a g^r)^s \bmod p$
$t_2 = t_1^{x_b} \bmod p$
$m = D_{hash(t_2)}(c)$,
then checks whether $r = hash(m, t_1)$.

Bob may pass (m, r, s) to others, who can be convinced that it indeed came from Alice by verifying

$$r = hash(m, (y_a g^r)^s)$$

4 Discussion

Discussion on Computation Cost

In the comparison of computation costs, we assume that the exponential computation is the most time consuming while the computation time for *hash* and (E, D) can be ignored. Under this assumption, the modified scheme requires a total of 5 exponential computations instead of 6 as in the traditional signature-then-encryption method. At the same time, it achieves exactly the same effect as in signature-then-encryption. That is, after receiving (c, r, s), Bob first obtains the plaintext message m using his private key. Now Bob has Alice's signature (m, r, s). This signature can be verified by anyone using only Alice's public key.

Discussion on Security

The security of the modified scheme is the same as that of the original scheme.

1. Unforgeability – it is computationally infeasible for an adaptive attacker to masquerade in creating a signcrypted text.

2. Non-repudiation – it is computationally feasible for a third party to settle a dispute between Alice and Bob in an event where Alice denies that she is the originator of a signcrypted text.

Both the unforgeability and non-repudiation are based on the assumption that it is computationally infeasible to forge (m, r, s)(without knowing x_a) such that

$r = hash(m, (y_a g^r)^s)$, which is the security basis for SDSS1. It should be noted that the same assumption is made in the Schnorr signature scheme [3].

3. Confidentiality — It is computationally infeasible for an adaptive attacker to gain any information on the contents of a signcrypted text. Here, as in any public key cryptosystems, we assume that $|m|$ is large enough to resist brute force attack from $r = hash(m, (y_a g^r)^s)$.

References

1. T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms", IEEE Transactions on Information Theory, IT-31(4):469-472, 1985.
2. H. Petersen and M. Michels, "Cryptanalysis and improvement of signcryption schemes", to appear in IEE Computers and Digital Techniques, 1998.
3. C. P. Schnorr, "Efficient identification and signature for smart cards", Advances in Cryptology - CRYPTO'89, LNCS 435, Springer-Verlag, pp. 239-251.
4. Y. Zheng, "Digital signcryption or how to achieve cost (signature and encryption) ¡¡ cost(signature) + cost(encryption)", In Advances in Cryptology - CRYPTO'97, LNCS 1294, Springer-Verlag, pp. 165-179, 1997.
5. Y. Zheng, "Signcryption and its application in efficient public key solutions", Pre-Proceedings of Information Security Workshop(ISW'97), pp. 201-218, to be published in LNCS by Springer-Verlag.

This article was processed using the LaTeX macro package with LLNCS style