An Improved Strong Designated Verifier Signcryption Scheme Postprint

Authors: Li Yuanxiao, Zhou Yanwei, Yang Bo

Date: 2018-12-13T00:00:00+00:00

Abstract

In 2012, Sujata et al. proposed a strong designated verifier signcryption scheme based on discrete logarithms; however, analysis reveals that Sujata et al.' s scheme cannot resist delegation attacks and the verification right is delegatable. To address the aforementioned shortcomings, an improved strong designated verifier signcryption scheme is presented, wherein only the designated verifier can verify the validity of the signcrypted ciphertext; furthermore, the designated verifier can generate a signcryption copy indistinguishable from the original signcrypted ciphertext. Security analysis demonstrates that the scheme can not only resist adaptive chosen-plaintext attacks, but also ensure the unforgeability of the signcrypted ciphertext while providing authentication. Due to these superior properties of the scheme, it has broad application prospects in real-world scenarios, such as blockchain, electronic voting, electronic tendering, and other similar contexts.

Full Text

Preamble

Vol. 37 No. 2
Application Research of Computers
ChinaXiv Cooperative Journal

An Improved Strong Designated Verifier Signcryption Scheme

Li Yuanxiao[1,2], Zhou Yanwei[1,2], Yang Bo[1,2]†

(1. School of Computer Science, Shaanxi Normal University, Xi' an 710119, China;
2. State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China)

Abstract: In 2012, Sujata et al. proposed a strong designated verifier signcryption scheme based on discrete logarithms. However, analysis reveals that Sujata et al.' s scheme cannot resist delegable attacks, and the verification right is delegable. To address these shortcomings, this paper presents an improved strong designated verifier signcryption scheme where only the designated verifier can verify the validity of the signcrypted ciphertext. Furthermore, the designated verifier can generate a signcryption transcript that is indistinguishable from the original signcrypted ciphertext.

Security analysis demonstrates that the proposed scheme not only resists adaptive chosen-plaintext attacks but also guarantees the unforgeability of the signcrypted ciphertext while providing authentication. Due to these superior performance characteristics, the scheme has broad application prospects in real-world scenarios such as blockchain, electronic voting, and electronic tendering.

Keywords: strong designated verifier signcryption; delegable attack; unforgeable; indistinguishable

0 Introduction

Signcryption, first proposed by Zheng in 1997, is a novel cryptographic primitive that simultaneously fulfills both digital signature and public-key encryption functions within a single logical step, offering lower computational and communication costs than traditional "sign-then-encrypt" approaches. In signcryption schemes, the sender typically generates a session key using the receiver' s public key to achieve symmetric encryption between the parties, while the receiver can produce an identical session key using their private key. The security of this session key underpins the security of authentication and encryption.

To enhance signer privacy, Jakobsson et al. introduced the concept of Strong Designated Verifier Signature (SDVS), where only the designated verifier believes in the signature' s validity. However, anyone obtaining the signature can verify its validity and determine that the true signer is one of two parties. To address this issue, Saeednia et al. proposed a strong designated verifier signature scheme in 2003 where only the holder of the designated verifier' s private key can verify the signature' s validity. Concurrently, Saeednia et al. first introduced the concept of strong designated verifier signcryption, which combines signcryption with designated verifier properties to achieve deniable authentication while ensuring message confidentiality.

In 2005, Lipmaa et al. proposed an attack model against designated verifier signatures called the delegable attack. They introduced a new security property for designated verifier signatures (termed non-delegatability) and proved that several existing designated verifier signature schemes could not resist delegable attacks. The fundamental idea of delegable attacks is that the signer or designated verifier can delegate signing or verification rights to a third party without

revealing their private key—a property undesirable in secure designated verifier signature schemes.

In 2012, Sujata et al. proposed a strong designated verifier signcryption scheme based on discrete logarithms. However, analysis shows that Sujata et al.' s scheme cannot resist delegable attacks and possesses delegable verification rights. Therefore, this paper presents an improved strong designated verifier signcryption scheme. In this scheme, signcrypted text can only be verified by the designated verifier, achieving significantly higher computational efficiency than the schemes in references [6,7] while providing the same functionality, along with security and efficiency analysis.

1.1 Discrete Logarithm Problem

Let p be a prime number and g be a primitive root of p, meaning all values of g generate the set $\{1, 2, ..., p-1\}$. For any $b \in \{1, 2, ..., p-1\}$, there exists a unique i such that $b \equiv g^i \pmod{p}$. This i is called the discrete logarithm of b with base g modulo p, denoted as $i = \log_g b \pmod{p}$. When g, p, and i are known, b can be computed easily using fast exponentiation algorithms. However, when g, p, and b are known, computing i is extremely difficult. Currently, the fastest known algorithm for solving discrete logarithms has time complexity $O(\exp((\ln p)^{1/3} \cdot (\ln \ln p)^{2/3}))$, making it infeasible when p is sufficiently large.

1.2.1 Indistinguishability

For any polynomial-time adversary $\mathcal{A}$, if there exists a negligible function $\varepsilon(k)$ such that $Adv_{\mathcal{A}}^{IND-CPA}(k) \leq \varepsilon(k)$, where k is the security parameter, then the algorithm is semantically secure, or equivalently, possesses indistinguishability under chosen-plaintext attacks (IND-CPA). As shown in Figure 1 [Figure 1: see original paper], the game proceeds as follows:

a) The challenger runs the key generation algorithm to generate the signcryption recipient' s public/private key pair (bpk, bsk) and sends bpk to adversary $\mathcal{A}$.

b) When adversary $\mathcal{A}$ makes signcryption queries, it selects a message $m \in \mathcal{M}$ and the signer' s public key apk, sending them to the challenger. The challenger simulates the signcryption oracle, performs signcryption, and returns the result to $\mathcal{A}$.

c) Adversary $\mathcal{A}$ selects two messages $m_0, m_1 \in \mathcal{M}$ of equal length and a public key bpk, sending them to the challenger. The challenger randomly selects $r \in \{0, 1\}$ and computes a signcryption δ^* for message m_r using the signer' s private key ask and the designated verifier' s public key bpk,

sending δ^* to $\mathcal{A}$ as the challenge ciphertext.

d) Adversary $\mathcal{A}$ outputs r'. If $r' = r$, the adversary succeeds in distinguishing. The adversary' s advantage is defined as the function of parameter k:

$$Adv_{\mathcal{A}}^{CPA}(k) = |Pr[r' = r] - \frac{1}{2}|$$

1.2.2 Unforgeability

For any polynomial-time adversary $\mathcal{A}$, if there exists a negligible function $\varepsilon(k)$ such that $Pr[Sig\text{-}forge_{\mathcal{A}}(n) = 1] \leq \varepsilon(n)$, then the signcryption scheme is existentially unforgeable under adaptive chosen-message attacks (EF-CMA). As shown in Figure 2 [Figure 2: see original paper], the game proceeds as follows:

a) The challenger runs the key generation algorithm to generate the signcryption recipient' s public/private key pair (bpk, bsk) and sends bpk to adversary $\mathcal{A}$.

b) When adversary $\mathcal{A}$ makes signcryption queries, it selects a message $m \in \mathcal{M}$ and the signer' s public key apk, sending them to the challenger. The challenger simulates the signing oracle, performs signing, and returns the result to $\mathcal{A}$.

c) Adversary $\mathcal{A}$ selects a message $m' \in \mathcal{M}$ and public key bpk, sending them to the challenger. The challenger computes a signcryption text δ' for message m' using the signer' s private key ask and the designated verifier's public key bpk, sending δ' to $\mathcal{A}$ as the challenge ciphertext.

d) Adversary $\mathcal{A}$ outputs (m^*, δ^*) such that $Vrfy(bpk, m^*, \delta^*) = 1$ and m^* was not previously queried. If these conditions hold, the adversary succeeds in forgery.

2.1 Scheme Overview

The scheme first generates public parameters: two safe primes p and q (where q is a prime factor of $p-1$), a generator g of order q, and a one-way hash function H whose output belongs to $\mathbb{Z}_p^*$.

Key Generation:
- Signer Alice selects her private key $x_a \in \mathbb{Z}_q^*$ and computes her public key $y_a = g^{x_a} \pmod{p}$.
- Designated verifier Bob similarly has key pair (x_b, y_b) where $x_b \in \mathbb{Z}_q^*$ and $y_b = g^{x_b} \pmod{p}$.

Signcryption Generation:
a) Alice randomly selects $k \in \mathbb{Z}_q^*$, $r, s, t \in \mathbb{Z}_q^*$.
b) Alice computes $K = g^k \pmod p$, $C = E_K(m)$, and $r = H(K)$.
c) Alice computes $s \equiv k - x_a \cdot r \pmod q$.
d) Alice sends the signcryption text $\delta = (r, s, C)$ to designated recipient Bob.

Unsigncryption:
a) Upon receiving $\delta = (r, s, C)$, Bob uses his private key x_b to compute $K' = (g^s \cdot y_a^r)^{x_b} \pmod p$.
b) Bob computes $r' = H(K')$ and verifies if $r' = r$. If the equality holds, the signature is valid.

2.2 Scheme Analysis

The original intention of designated verifier signatures is that Alice proves the correctness of statement Θ to Bob by demonstrating either "Θ is correct" or "she knows Bob' s private key." However, in this designated verifier signature scheme design, the proof transforms into demonstrating the proposition "Θ is correct" or "she knows partial information," where this partial information is a one-way function of Alice' s private key (public key) and Bob' s public key (private key)—meaning the private key cannot be derived from this one-way function. In cryptography, except for private keys which are strictly protected, all other components are vulnerable under various attacks and may be compromised.

This scheme possesses the delegatability flaw, described as follows: Because any third party obtaining (r, s, C) can verify whether $g^k \equiv (g^s \cdot y_a^r)^{x_b} \pmod p$ holds without knowing Bob' s private key, this violates the definition of strong designated verifiers. After obtaining knowledge [8], anyone can verify whether the equation holds without the designated verifier Bob' s private key. An attacker, upon acquiring this knowledge before the message reaches the designated verifier, can decrypt the ciphertext on one hand and learn the true origin of the signcryption on the other—an extreme security risk for strong designated verifier signcryption.

---

3 Construction of the Proposed Scheme

This paper presents a new strong designated verifier signcryption scheme that resists delegable attacks and offers certain security improvements over Sujata' s signcryption scheme. Based on Lee's strong designated verifier signature scheme [9], this work combines Schnorr signatures [10] with Wang' s authenticated encryption scheme [11], using a secure symmetric encryption algorithm.

a) System Initialization:
Two safe primes p and q (where q is a prime factor of $p - 1$), a generator g of

order q, and a one-way hash function H whose output belongs to $\mathbb{Z}_p^*$.

b) Key Generation:
- Alice selects her private key $x_a \in \mathbb{Z}_q^*$ and computes her public key $y_a = g^{x_a} \pmod p$.
- Designated verifier Bob similarly has key pair (x_b, y_b) where $x_b \in \mathbb{Z}_q^*$ and $y_b = g^{x_b} \pmod p$.

c) Signcryption Generation:
(a) Alice selects a random value $k \in \mathbb{Z}_q^*$.
(b) Alice computes $r = g^k \pmod p$ and $s \equiv k + x_a \cdot H(m, r) \pmod q$.
(c) Alice splits c bitwise into left and right halves c_1, c_2 (if c length is odd, left half is shorter).
(d) Alice computes $t = H(m, c_1)$ and $D = Enc_{c_2}(m)$.
(e) The generated signcryption text is $\delta = (r, t, D)$.

d) Unsigncryption:
(a) Upon receiving $\delta = (r, t, D)$, Bob uses his private key x_b to compute $c' = (y_a \cdot g^r)^{x_b} \pmod p$.
(b) Bob splits c' into two parts c_1', c_2' and computes $m' = Dec_{c_2'}(D)$.
(c) If $t = H(m', c_1')$ holds, the signcryption text is verified.

e) Signcryption Transcript Generation:
The characteristic of designated verifier signatures is that only the pre-designated verifier knows the true origin of the signature and cannot convince any third party of its authenticity. The reason third parties cannot know the true origin is that the designated verifier can also generate a signature transcript indistinguishable from the signer' s signature. To make it difficult for third parties to distinguish whether the transmitted signcryption text was generated by Alice or Bob, after successful verification, Bob simulates the generation of a signcryption transcript δ' as follows:

(a) Bob selects a random value $k' \in \mathbb{Z}_q^*$.

(b) Bob computes $r' = g^{k'} \pmod p$ and $s' \equiv k' + x_b \cdot H(m, r') \pmod q$.

(c) Using the same method, Bob splits c' into two parts c_1', c_2'.

(d) Bob generates the signcryption transcript as $\delta' = (r', t', D')$.

---

4 Security and Efficiency Analysis

In this scheme, given only the public keys of the signcrypter and designated verifier along with knowledge of (r, t, D), one cannot generate a valid signature or verify its validity. Even with this knowledge, generating a valid signature or verifying its validity still requires solving the discrete logarithm problem.

This scheme possesses unforgeability, confidentiality, non-transferability, and non-delegatability.

4.1 Correctness

Recipient Bob can correctly verify the signature' s validity upon receiving signcryption text $\delta = (r, t, D)$ because:

$$c' = (y_a \cdot g^r)^{x_b} \equiv g^{x_a x_b} \cdot g^{r x_b} \equiv g^{k x_b} \pmod{p}$$

Thus, the verification equation $t = H(m', c_1')$ holds.

4.2 Confidentiality

Except for the designated recipient, no one can extract any information about message m from the signcryption text. Due to the one-way property of hash functions, recovering c_2 from D cannot yield any information about message m. The pair $(Enc(\cdot), Dec(\cdot))$ must be a secure symmetric encryption/decryption algorithm. To obtain the key, one must know k and x_b. Due to the difficulty of the discrete logarithm problem, it is hard to derive k and x_b from c, making the scheme secure.

Lemma 1: For all probabilistic polynomial-time adversaries $\mathcal{A}$, there exists a negligible function $\varepsilon(k)$ satisfying $Pr[PrivK^{cpa}_{\mathcal{A},\Pi}(n) = 1] \leq \frac{1}{2} + \varepsilon(n)$, meaning the scheme is indistinguishable under chosen-plaintext attacks.

In this scheme, if adversary $\mathcal{A}$ successfully distinguishes between m_0, m_1, it implies $\mathcal{A}$ has broken the secure symmetric encryption scheme. Due to the ciphertext indistinguishability of symmetric encryption, this scheme is indistinguishable. The symmetric encryption component can be constructed using alternative methods with provable security, but for scheme completeness, detailed description is omitted here.

4.3 Unforgeability

The scheme uses Schnorr signatures to generate (r, s). Based on the provable security of Schnorr signatures [4], no adaptive adversary, including Bob, can forge a signcryption text for message m satisfying $s \equiv k + x_a \cdot H(m, r) \pmod{q}$; otherwise, the adversary could successfully forge a valid Schnorr signature. Therefore, this scheme is secure.

Lemma 2: If no probabilistic polynomial-time algorithm can win the above game with non-negligible probability, the scheme is existentially unforgeable under adaptive chosen-message attacks.

If, in the above game, the adversary returns a valid message-signcryption text to the challenger after a series of queries, then the adversary' s ability to forge

signcryptions can be reduced to breaking the Schnorr signature problem. Since Schnorr signatures are unforgeable, this scheme is unforgeable.

4.4 Non-delegatability

Given only the public keys of the signcrypter and designated verifier along with knowledge of (r, t, D), one cannot generate a valid signature or verify its validity. This scheme eliminates delegatability flaws in the verification process—only those possessing the designated verifier' s private key can verify signature validity, which requires solving the discrete logarithm problem, a computationally difficult problem with current computing capabilities.

4.5 Non-transferability

The signcryption transcript δ' generated by Bob is indistinguishable from Alice' s signcryption δ. From Alice' s valid signcryption text, randomly selecting (r, t, D) and (r', t', D'), where r is determined by k and r' is determined by k', the distributions of (r, t, D) and (r', t', D') are identical and therefore indistinguishable.

Thus, Bob' s generated signcryption transcript $\delta' = (r', t', D')$ is indistinguishable from Alice' s signcryption $\delta = (r, t, D)$.

4.6 Efficiency Analysis

In this scheme, the computational complexity of the signcryption phase is $2T_E + T_H + T_M$, and the unsigncryption phase is $2T_E + T_H + T_M$ (where T_E, T_H, T_M, T_B represent modular exponentiation, hash operation, modular multiplication, and bilinear pairing operations respectively). Compared with reference [2], this scheme achieves delegatable attack resistance without additional overhead. Furthermore, Table 1 compares the computational efficiency of signcryption schemes with equivalent functionality.

Table 1 Performance comparison of relevant schemes

To demonstrate the scheme' s actual computational overhead and corresponding time-cost curves, this paper implemented the scheme in C++ using the Crypto++ library in the Visual Studio 2012 integrated development environment. Table 2 shows the main parameters of the experimental environment.

Table 2 Main parameters of the experimental environment

When the private keys of the signcrypter and designated verifier are fixed (in this experiment, $x_a = x_b = 512$), the runtime curves for signcryption and unsigncryption processes with varying message lengths are shown in Figure 3 [Figure 3: see original paper].

Figure 3 Runtime curves for signcryption and unsigncryption processes

Through theoretical and implementation analysis, we conclude that this scheme is efficient and possesses practical application value.

---

5 Conclusion

This paper presents a delegable attack on Sujata' s scheme and proposes a strong designated verifier signcryption scheme based on the discrete logarithm problem. This scheme enables secret transmission between the signcrypter and designated verifier without requiring a secure channel [15]. The scheme is proven secure under adaptive chosen-plaintext attacks. Additionally, it satisfies the non-delegatability security requirement and prevents ciphertext tampering during signcryption transmission [16]. Therefore, this scheme has numerous practical applications in areas such as blockchain asset proofs [17,18] and electronic voting.

References

[1] Zheng Yuliang. Digital signcryption or how to achieve cost (signature&encryption) « cost (signature) + cost (encryption) [C]// Advances in Cryptology. Berlin: Springer, 1997: 165-179.

[2] Jakobsson M, Sako K, Impagliazzo R. Designated verifier proofs and their applications [C]// Advances in Cryptology. Berlin: Springer, 1996: 143-154.

[3] Saeednia S, Kremer S, Markowitch O. An efficient strong designated verifier signature scheme [C]// Proc of International Conference on Information Security and Cryptology. Berlin: Springer, 2004: 40-54.

[4] Lipmaa H, Wang Guilin, Bao Feng. Designated verifier signature schemes: attacks, new security notions and a new construction [C]// Proc of the 32nd International Colloquium-ICALP2005. Berlin Heidelberg: Springer, 2005: 459-471.

[5] Sujata Mohanty, Banshidhar Majhi. A Strong Designated Verifiable DL Based Signcryption Scheme [J]. Journal of Information Processing Systems, 2012, 8(4): 567-574.

[6] Tan C H. Analysis of improved signcryption scheme with key privacy [J]. Information Processing Letters, 2006, 99(4): 135-138.

[7] Huang Qiong, Willy S, Wong D S. Non-delegatable Identity-based Designated Verifier Signature [R]. Cryptology ePrint Archive: Report 2009/315.

[8] Yang Xiaoyuan, Yu Qingfei. ECC-based new designated verifier signature scheme [J]. Journal of PLA University of Science and Technology, 2007, 35(8): 1432-1436.

[9] Lee J, Chang J. Comment on Saeednia et al.' s strong designated verifier signature scheme [J]. Computer Standards & Interfaces, 2009, 31(1): 258-260.

[10] Schnorr C P. Efficient signature generation for smart cards [J]. Journal of Cryptology, 1991, 4(3): 239-252.

[11] Wang Guilin, Bao Feng, Ma Changshe, et al. Efficient authenticated encryption schemes with public verifiability [C]// Proc of the 60th IEEE Vehicular Technology Conference on Wireless Technologies for Global Security. Washington DC: IEEE Computer Society, 2004: 3258-3261.

[12] Petersen H, Michels M. Cryptanalysis and improvement of signcryption schemes [J]. IEEE Computers and Digital Communications, 1998, 8(2): 123-135.

[13] Yang Guomin, Wong D S, Deng Xiaotie. Analysis and improvement of Petersen-Michels signcryption scheme with key privacy [C]// Proc of Information Security Conference. Singapore: Springer, 2005: 218-232.

[14] Huang Qiong, Yang Guomin, Wong D S, et al. Identity-based strong designated verifier signature revisited [J]. Journal of Systems and Software, 2011, 84(1): 120-129.

[15] Yang Bo, Sun Ying, Yu Yong, et al. A Strong designated verifier signature scheme with secure disavowability [C]// Proc of International Conference on Intelligent Networking & Collaborative Systems. Bucharest, Romania: IEEE SMC, 2012: 286-291.

[16] Yang Bo, Yu Yong, Sun Ying. A novel construction of SDVS with secure disavowability [J]. Cluster Computing, 2013, 16(4): 807-815.

[17] Koochak S M, Ahmadian-Attari M, Aref M R. Provably secure strong designated verifier signature scheme based on coding theory [J]. International Journal of Communication Systems, 2016, 30(7): e3124.

[18] Wang Huaqun, He Debiao, Ji Yimu. Designated-verifier proof of assets for bitcoin exchange using elliptic curve cryptography [J]. Future Generation Computer Systems, 2017, 84(1): 135-137.

Note: Figure translations are in progress. See original paper for figures.