Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.12785/amis/090250

Data Protection for Computer Forensics using Cryptographic Mechanisms

*Tzong-Sun Wu and Han-Yu Lin**

Department of Computer Science and Engineering, National Taiwan Ocean University, Keelung, 202, Taiwan

Received: 21 Jun. 2014, Revised: 19 Sep. 2014, Accepted: 21 Sep. 2014
Published online: 1 Mar. 2015

Abstract: Data protection is an important issue for computer forensics in a digitalized world. We construct secure data protection methods by adopting cryptographic building blocks. In our proposed scheme, an investigating authority delegates a group composed of n judicial policemen to collect digital evidence. Any t or more of them can cooperatively generate valid authenticated evidence for collected ordinary evidence on behalf of the investigating authority. To ensure confidentiality, the authenticated evidence can only be decrypted and verified by a designated investigator of Investigation of Bureau, Ministry of Justice (MJIB). For the litigation process, the designated investigator is capable of further converting the authenticated evidence into an ordinary one and giving it to a judge or prosecutor without leaking the information of his private key. We also present a variant with message linkages for benefiting the encryption of a large message. To guarantee the feasibility of practical implementation, we show that our construction achieves the IND-CCA2 and the EF-CMA security in the random oracle model.

Keywords: Data protection, computer forensics, cryptographic mechanism, random oracle, security.

1. Introduction

Data protection has always been an important issue in either a real or a digitalized world. Computer forensics is a forensic science which aims for explaining and preserving the current state of data, i.e., digital evidence, collected from computers and digital storage media. The field of computer forensics also includes firewall forensics, network forensics, database forensics and mobile device forensics [33]. When transmitting the digital evidence via an insecure channel like the Internet, we must pay special attention to protect these data from being eavesdropped or unauthorized modification. In realistic legal cases, an investigating authority may delegate a team of several judicial policemen to conduct forensic processes and collect digital evidence. When sufficient judicial policemen confirm the state of obtained evidence, they can cooperatively sign on behalf of the investigating authority and deliver the authenticated evidence to a designated investigator of Investigation Bureau, Ministry of Justice (MJIB). For the confidentiality concern, only the designated investigator is able to decrypt and verify the authenticated evidence. He can also reveal the converted ordinary evidence to a judge or prosecutor for the litigation process. A diagram of the procedures for computer forensics in legal cases is illustrated as Fig. 1.

It is believed that cryptographic mechanisms can fulfill above mentioned application for computer forensics. In 1976, Diffie and Hellman [5] introduced the first public key system. In a public key system, everyone owns a private key together with its corresponding public one such that he can either perform public key encryptions [11] or generate digital signature [6,22]. The former guarantees confidentiality [8] while the latter ensures authenticity [12,19,24] and non-repudiation [18]. In 1979, Shamir [14] came up with a (t, n) threshold secret sharing scheme in which a master secret is divided into n secret shares and stored by different users. Any t or more users can cooperatively reconstruct the master secret while less than or equal to $t-1$ cannot. To meet more diversified application requirements, Mambo $et\ al.$ [15,16] proposed proxy signature schemes in 1996. In a proxy signature scheme, an authorized person called proxy signer can legitimately produce proxy signatures on behalf of an original signer. As to further supporting group-oriented applications, some researchers [2,9,10,13,26,31,32,34] have also devoted their attention to the design of proxy signature variations.

* Corresponding author e-mail: lin.hanyu@msa.hinet.net

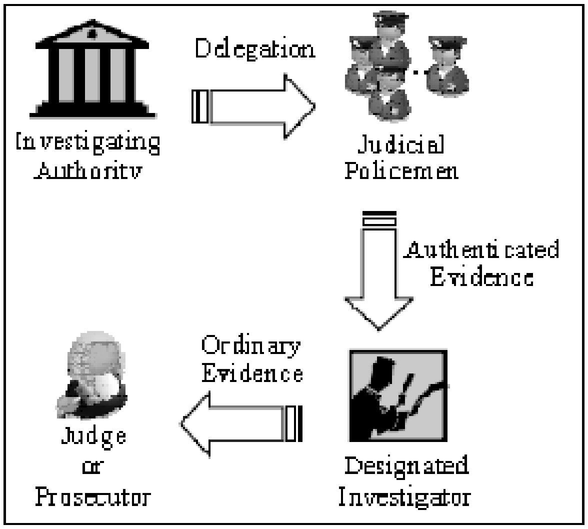

Figure 1 Diagram of the procedures for computer forensics.

In 1994, Horster *et al.* [7] proposed an authenticated encryption (AE) scheme which simultaneously satisfies the properties of confidentiality and authenticity. Such schemes allow a signer to generate an authenticated ciphertext and only the designated recipient has the ability to recover the message and verify its corresponding signature. In 1997, Zheng [35] addressed a signcryption scheme which also serves the same functionalities as AE schemes. To prevent a dishonest signer from repudiating his generated ciphertext, in 1999, Araki *et al.* [1] proposed a convertible limited verifier signature scheme with a later arbitration mechanism. Yet, if a dishonest signer refuses to assist, the mechanism is infeasible. In 2002, Wu and Hsu [28] introduced a convertible authenticated encryption (CAE) scheme in which the designated recipient can solely prove the signer's dishonesty in case of a repudiation dispute. Due to the limited system bandwidth, it is often difficult to encrypt a large message. In 2005, Peng *et al.* [20] addressed a publicly verifiable AE scheme with message linkages for transmitting a large message. Later, Lv *et al.* [14] also proposed a more practical one for realistic implementation.

Team work is an important approach in an organization to promote the efficiency. Sometimes, it can also be adopted to escalate the security level. In 2008, Wu *et al.* [29] proposed a group-oriented CAE scheme allowing multiple signers to cooperatively generate a valid authenticated ciphertext. In 2009, Tsai [25] presented another variant with better efficiency. However, Tsai's scheme cannot assure the property of confidentiality. Based on Wu *et al.*'s scheme, Chang [3] addressed a different scheme with shared verification of multiple designated recipients. For facilitating the operation of proxy delegation, Wu and Lin [30] proposed a proxy CAE scheme which enables a group consisting of n original signers to cooperatively delegate their signing power to a proxy signer who can therefore generate a valid authenticated ciphertext on behalf of the original group. Note that in the Wu-Lin scheme, all the original signers must agree on the proxy delegation.

According to the diagram depicted in Figure 1, we can observe that none of the above single existing cryptographic mechanism perfectly solves the realistic application requirement for computer forensics. It thus can be seen that the design of secure and feasible method fulfilling the requirement from the perspective of realistic consideration is crucial and benefits to the practical applicability.

2. Formal model of our scheme

We describe the formal model of our proposed scheme including involved parties and algorithms in this section. The used notations are defined as Table 1.

Table 1 The used notations

N	set of all natural numbers
Z_p	integers modulo p
Z_p^*	multiplicative group of integers modulo p
$GF(p)$	Galois field of p elements
$x \in Z_p$	element x in set Z_p
$x \in_R Z_p$	element x is a random integer in set Z_p
$S \backslash T$	difference of sets S and T
$\#Z_p$	number of elements in set Z_p
$x \longleftarrow Z_p$	sampling element x uniformly in set Z_p
a mod b	modulo operation: reminder of a divided by b
$a \mid b$	integer b is divisible by integer a
$a \parallel b$	concatenation of a and b
$\|x\|$	bit-length of integer x, also absolute value of x
$\sum_{i=1}^{n} v_i, \sum_{i \in S} v_i$	sum of values v_i for i = 1, 2, ..., n, or for $i \in S$
$\prod_{i=1}^{n} v_i, \prod_{i \in S} v_i$	product of values v_i for i = 1, 2, ..., n, or for $i \in S$
$\log_b x$	logarithm to base b of x
$\oplus$	logical operation XOR
$\neg$	logical operation NOT
$\wedge$	logical operation AND
$\vee$	logical operation OR
$\forall$	for all
$Pr[E]$	probability of event E occurring

2.1. Involved parties

There are three major involved parties: an investigating authority, a group of n judicial policemen and a designated investigator of MJIB. Each one is a probabilistic polynomial-time Turing machine (PPTM) [17]. The investigating authority delegates judicial policemen to conduct forensic processes. Any t or more judicial policemen can cooperatively sign and produce valid authenticated evidence on behalf of the investigating authority while less than or equal to $t-1$ cannot. Finally, the designated investigator of MJIB verifies received authenticated evidence. He can also reveal converted ordinary evidence for the subsequent litigation process.

2.2. Algorithms

The proposed scheme consists of four algorithms. According to the procedures depicted in Fig. 1, when a forensic system is established, we have to run "Setup" algorithm to obtain system's public parameters. For the delegation process, an investigating authority can run "Proxy Credential Generation (PCG)" algorithm to delegate his power to a group of judicial policemen. To generate the authenticated evidence for a designated investigator of MJIB, judicial policemen have to run "Authenticated-Evidence-Generation (AEG)" algorithm. Upon receiving the authenticated evidence, the designated investigator can run "Authenticated Evidence Verification (AEV)" algorithm to validate the digital evidence. Details of the four algorithms are described as follows:

Setup: Taking as input 1^k where k is a security parameter, the algorithm generates the system's public parameters params.

Proxy-Credential-Generation (PCG): The PCG algorithm takes as input a warrant, the identity for the group of judicial policemen and the private key of investigating authority. It outputs the corresponding proxy credential which can be also regarded as a delegation agreement of the issuing authority.

Authenticated-Evidence-Generation (AEG): The AEG algorithm takes as input a proxy credential, evidence m, the public key of designated investigator of MJIB and the private key of the group for judicial policemen. It generates corresponding authenticated evidence δ.

Authenticated-Evidence-Verification (AEV): The AEV algorithm takes as input authenticated evidence δ, the private key of designated investigator of MJIB and the public keys of investigating authority and the group for judicial policemen. It outputs the converted ordinary evidence (m, Ω) if δ is valid. Otherwise, an error symbol ¶ is returned as a result.

3. The proposed scheme

In this section, we give the concrete construction of our scheme and its variant with message linkages.

3.1. Construction

Setup: Taking as input 1^k, the system authority (SA) selects a $t-1$ degree polynomial $f(x) = d_0 + d_1x + ... + d_{t-1}x^{t-1}$ with d_i's $\in Z_q$, two large primes (p, q) and a generator g of order q, where $|q| = k$ and $q|(p-1)$. Let h_1: $\{0,1\}^k \times Z_p^* \to Z_q$, h_2: $\{0,1\}^k \times Z_p^* \to Z_q$, h_3: $Z_p^* \to Z_q$ and h_4: $Z_p^* \to \{0,1\}^k$ be collision resistant hash functions which would never output the same result for different input values. The system announces public parameters params = $\{p, q, g, h_1, h_2, h_3\}$ and derives each party U_i's private key $x_i = f(i)$. The corresponding public key is computed as $y_i = g^{x_i} \bmod p$.

Proxy-Credential-Generation (PCG): Let U_o be an investigating authority delegating his power to a group of judicial policemen $PG = \{U_1, U_2, ..., U_n\}$. U_o first chooses a secret integer $t_0 \in_R Z_q$ to compute

$$T = g^{t_0} \bmod p, \tag{1}$$

$$\sigma = t_0 - x_o h_1(m_w, T) \bmod q, \tag{2}$$

where m_w is the warrant consisting of the identifier of investigating authority and the group for judicial policemen, the delegation duration and so on. Note that the proxy credential (σ, T) is regarded as the signature for m_w.

(σ, m_w, T) is then sent to PG via a secure channel which can prevent transmitted information from being intercepted and tampered. Upon receiving (σ, m_w, T), each $U_i \in PG$ can first compute C as Eq. (3) and then perform Eq. (4) to check its validity.

$$C = y_o^{h_1(m_w,T)} \bmod p, \tag{3}$$

$$T = g^{\sigma} C \pmod p. \tag{4}$$

If it does not hold, (σ, m_w, T) is requested to be sent again.

We show that the verification of Eq. (4) works correctly. From the right-hand side of Eq. (4), we have

$$\begin{aligned} & g^{\sigma} C \\ &= g^{\sigma} y_o^{h_1(m_w,T)} && \text{(by Eq. (3))} \\ &= g^{t_0 - x_o h_1(m_w,T)} y_o^{h_1(m_w,T)} && \text{(by Eq. (2))} \\ &= g^{t_0} \\ &= T \pmod p && \text{(by Eq. (1))} \end{aligned}$$

which leads to the left-hand side of Eq. (4).

Authenticated-Evidence-Generation (AEG): Without loss of generality, let $SPG = \{U_1, U_2, ..., U_t\}$ be the subgroup composed of t judicial policemen who can cooperatively generate a valid authenticated evidence on behalf of the group PG, and U_{ck} a semi-trusted clerk who is responsible for verifying individual's authenticated evidence and combining them. A semi-trusted third party is said to be honest but curious, i.e., he will not perform anything that deviates from the predefined procedures, but he might attempt to learn any secret information from observed messages. The private key of PG is d_0 and the corresponding public key is $y_D = g^{d_0} \bmod p$. By using the Lagrange Interpolation [27], any t or more members of the group PG can cooperatively reconstruct the $t-1$ degree polynomial $f(x)$ with their key pairs and then derive the group private key $d_0 = f(0)$. To generate the authenticated evidence δ for obtained ordinary evidence m, each $U_i \in SPG$ first chooses an integer $r_i \in_R Z_q$ to compute

$$c_i = \prod_{U_j \in SPG \setminus \{U_i\}} j/(j-i) \bmod q, \quad (5)$$

where c_i is the Lagrange coefficient [27],

$$e_i = c_i \cdot x_i \bmod q, \quad (6)$$
$$R_i = g^{r_i} \bmod p, \quad (7)$$

and then sends R_i to $U_j \in SPG\setminus\{U_i\}$ and U_{ck}. Upon receiving all R_j's, each $U_i \in SPG$ computes

$$R = \prod_{j=1}^{t} R_j \bmod p, \quad (8)$$
$$s_i = r_i - e_i h_2(m, C, R) \bmod q. \quad (9)$$

s_i is then delivered to the clerk U_{ck}. After receiving all s_i's, U_{ck} verifies if

$$R_i = g^{s_i} y_i^{c_i h_2(m,C,R)} \bmod p. \quad (10)$$

If it does not hold, s_i is requested to be sent again; else, U_{ck} chooses $z \in_R Z_q$ to computes

$$S = \prod_{U_j \in SPG} s_j \bmod q, \quad (11)$$
$$K = y_v^{\sigma} \bmod p, \quad (12)$$
$$W = h_3(CK \bmod p) S \bmod q, \quad (13)$$
$$Q = h_4(K) \oplus m. \quad (14)$$

The authenticated evidence $\delta = (Q, W, R, T)$ and m_w are then delivered to the designated investigator of MJIB U_v.

Authenticated-Evidence-Verification (AEV): Upon receiving (δ, m_w), U_v first derives C as Eq. (3), computes

$$K = (TC^{-1})^{x_v} \bmod p, \quad (15)$$
$$S = h_3(CK \bmod p)^{-1} W \bmod q, \quad (16)$$
$$m = Q \oplus h_4(K), \quad (17)$$

and then checks the redundancy embedded in m. U_v can further verify the authenticated evidence by checking if

$$R = g^S y_D^{h_2(m,C,R)} \pmod p. \quad (18)$$

We demonstrate that the designated investigator of MJIB can recover the ordinary evidence m with its embedded redundancy by Eq. (17). From the right-hand side of Eq. (17), we have

$$\begin{aligned} & Q \oplus h_4(K) \\ &= Q \oplus h_4((TC^{-1})^{x_v} \bmod p) && \text{(by Eq. (15))} \\ &= Q \oplus h4((g^{\sigma})^{x_v} \bmod p) && \text{(by Eq. (4))} \\ &= Q \oplus h4(y_v^{\sigma} \bmod p) \\ &= m && \text{(by Eqs. (12) and (14))} \end{aligned}$$

which leads to the left-hand side of Eq. (17).

If the authenticated evidence (Q, W, R, T) is correctly generated, it will pass the test of Eq. (18). From the right-hand side of Eq. (18), we have

$$\begin{aligned} & g^S y_D^{h_2(m,C,R)} \\ &= g^{\sum_{U_j \in SPG} s_j} g^{d_0 h_2(m,C,R)} && \text{(by Eq. (11))} \\ &= g^{\sum_{U_j \in SPG} s_j + h_2(m,C,R) c_j x_j} \\ & && \text{(by Lagrange Interpolation [27])} \\ &= g^{\sum_{U_j \in SPG} s_j + h_2(m,C,K,R) e_j} && \text{(by Eq. (6))} \\ &= g^{\sum_{U_j \in SPG} r_j} && \text{(by Eq. (9))} \\ &= \prod_{j=1}^{t} R_j \\ &= R \pmod p && \text{(by Eq. (8))} \end{aligned}$$

which leads to the left-hand side of Eq. (18).

For the subsequent litigation process, the designated investigator of MJIB U_v can reveal converted ordinary evidence $(m, \Omega = (S, R, T))$ and the warrant m_w to a judge or a prosecutor who can therefore verify it with the assistance of Eq. (18). Note that the converted ordinary evidence has been derived during the previous forensic process. Consequently, it takes no extra computational efforts for U_v to conduct the conversion of ordinary evidence.

3.2. Variant with message linkages

Due to the limited system bandwidth, it often causes the difficulty in encrypting a large message. In the subsection, we slightly modify our proposed scheme to present its variant with message linkages. The construction is similar to that in Section 4.1. We only describe the different parts as follows:

Authenticated-Evidence-Generation (AEG): For generating the authenticated evidence for a large message m, each $U_i \in SPG$ first divides m into f pieces, i.e., $m = m_1 \parallel m_2 \parallel \ldots \parallel m_f$ such that each m_l has a suitable length, and then chooses $r_i \in_R Z_q$ to compute (c_i, e_i, R_i, R, s_i) as those in Section 3.1. The parameter Q_l is computed as

$$Q_l = m_l \cdot h_4(Q_{l-1} \oplus h_4(K)) \bmod p, \quad (14^*)$$

for $l = 1, 2, ..., f$, where $Q_0 = 0$. The authenticated evidence δ = (W, R, T, Q_1, Q_2, ..., Q_f) and m_w are then delivered to U_v.

Authenticated-Evidence-Verification (AEV): Upon receiving it, U_v first derives (C, K) as Eqs. (3) and (15), respectively. He then computes

$$m_l = Q_l h_4(Q_{l-1} \oplus h_4(K))^{-1} \bmod p, \quad \text{for } l = 1,2,...,f, \tag{17*}$$

and recovers the original m as $m_1 \parallel m_2 \parallel ... \parallel m_f$. U_v can further verify the authenticated evidence by checking Eq. (18).

We show that with $\delta = (W, R, T, Q_1, Q_2, ..., Q_f)$ and m_w, the designated investigator of MJIB U_v can recover m and check its validity with Eq. (17*). From the right-hand side of Eq. (17*), we have

$$\begin{aligned} & Q_l \cdot h_4(Q_{l-1} \oplus h_4(K))^{-1} \\ &= Q_l \cdot h_4(Q_{l-1} \oplus h_4(K)) \cdot h_4(Q_{l-1} \oplus h4(K))^{-1} \quad \text{(by Eq. (14*))} \\ &= m_l \pmod p \end{aligned}$$

which leads to the left-hand side of Eq. (17*).

3.3. Efficiency analyses

Table 2 summarizes the functionalities among the proposed and related works including Lv *et al.*'s (LW for short) [14], the Wu-Hsu (WH for short) [28], the Wu-Lin (WL for short) [30], Wu *et al.*'s (WT for short) [29], Chang's (Ch for short) [3] and Tsai's (Ts for short) [25].

Table 2 Comparisons in terms of functionalities

	LW	WH	WL	WT Ch	Ts	Ours
Group Oriented	N	N	Y	Y	Y	Y
Threshold Mechanism	N	N	N	N	N	Y
Proxy Delegation	N	N	Y	N	N	Y
Message Linkage	Y	N	N	N	N	Y
Conversion (of Evidence)	Y	Y	Y	Y	Y	Y
Conversion Free	Y	Y	Y	Y	Y	Y
Proof of Confidentiality	N	N	Y	Y	N	Y
Proof of Unforgeability	N	N	Y	Y	N	Y

Remark: For each compared scheme, "Y" and "N" separately denote "with" and "without" the evaluated functionality.

Tables 3 and 4 summarize the computational costs in number of the most time-consuming operation, i.e., modular exponentiation, among the proposed and above group-oriented schemes [3,25,29,30]. Note that Table 3 further evaluates the communication overheads between ours and WL [30], since their work also provides the functionality of proxy delegation.

Table 3 Comparisons of group-oriented schemes without the functionality of proxy delegation

	Encryption & Verification (excluding delegation)
WT	$4n^2 - 2n + 5$
Ch	$4n^2 - 2n + 5$
Ts	$3n + 5$
Ours	$3t + 5$

Remark: The parameter t is a threshold value and $t \leq n$.

Table 4 Comparisons of group-oriented scheme with the functionality of proxy delegation

	WL	Ours
Encryption & Verification (including delegation)	≈ 9.183	≈ 9.171
Communication Overheads (including authenticated & ordinary evidence)	$5\|p\| + 4\|q\|$ ≈ 4608 bits	$4\|p\| + 4\|q\|$ ≈ 4096 bits

Remark: To obtain fair comparison results under the same basis, we consider single-user setting and let $|p| = |q|$ = 512 bits in both evaluated schemes.

4. Security proof

In this section, we address the security model with respect to the proposed scheme and then give detailed security proofs. Some necessary cryptographic security notions [4] are briefly reviewed as follows:

Discrete Logarithm Problem; DLP

Let p and q be large primes satisfying $q|(p-1)$, and g a generator of order q over $GF(p)$. The discrete logarithm problem is, given an instance (y, p, q, g), where $y = g^x$ mod p for some $x \in Z_q$, to derive $x = log_{p,q,g}y$.

Computational Diffie-Hellman Problem; CDHP

Let p and q be large primes satisfying $q|(p-1)$, and g a generator of order q over $GF(p)$. The computational Diffie-Hellman problem is, given an instance (p, q, g, g^a, g^b) for some $a, b \in Z_q$, to derive g^{ab} mod p.

4.1. Security model

Any cryptographic scheme simultaneously satisfying the properties of confidentiality and authenticity should consider the security requirements of message confidentiality and unforgeability. The widely accepted notion for the security of message confidentiality comes from the definition of indistinguishability- based security, i.e., the adversary attempts to distinguish a target ciphertext with respect to two candidate messages. In the taxonomy of cryptanalysis, there are three kinds of attacks: ciphertext-only attack, chosen-ciphertext attack (CCA) and adaptive chosen-ciphertext attack (CCA2). An adversary in ciphertext-only attack cannot make any query while that in CCA can query the plaintext for his chosen ciphertext once. An adversary in CCA2 is the most advantageous since he can adaptively make new queries based on previous results. We therefore consider an adversary in CCA2 against our proposed scheme in the security requirement of message confidentiality. In addition to the AEV queries, we also give the adversary the ability to make PCG and AEG queries. When it comes to the security requirement of unforgeability, we usually refer to an adversary in adaptive chosen-message attack (CMA). Such an adversary attempts to forge a valid authenticated ciphertext for his chosen message and is permitted to adaptively make PCG and AEG queries in our defined security notion. We design two game models for the above two crucial security requirements as Definitions 1 and 2, respectively.

Then we can formally prove the security of our scheme in the random oracle model. Namely, the one-way hash function is simulated as a random oracle controlled by a challenger who is responsible for answering the adversary's queries in the defined game model. Note that the simulated results of each random query should be computationally indistinguishable from those generated by a real scheme. Basically, the concept of security proof is a security reduction. That is to say, we can reduce a well-known cryptographic problem such as CDHP to our proposed scheme meaning that if there is any adversary winning the game in CCA2 or CMA, the challenger that takes the adversary's advantages is able to break CDHP.

Definition 1. (Confidentiality) *A cryptographic scheme is said to achieve the security requirement of confidentiality against indistinguishability under adaptive chosen-ciphertext attacks (IND-CCA2) if there is no probabilistic polynomial-time adversary $\mathcal{A}$ with non-negligible advantage in the following game played with a challenger $\mathcal{B}$:*

Setup: The challenger $\mathcal{B}$ first runs the Setup(1^k) algorithm and sends the system's public parameters $params$ to the adversary $\mathcal{A}$.

Phase 1: The adversary $\mathcal{A}$ can make several queries adaptively, i.e., each query might be based on the result of previous queries:

- *Proxy-Credential-Generation (PCG) queries:* $\mathcal{A}$ makes a PCG query with respect to the identity of target group for judicial policemen. $\mathcal{B}$ returns the corresponding proxy credential along with its warrant.
- *Authenticated-Evidence-Generation (AEG) queries:* $\mathcal{A}$ first chooses an ordinary evidence m and then gives it to $\mathcal{B}$ who will return corresponding authenticated evidence δ with a warrant m_w.
- *Authenticated-Evidence-Verification (AEV) queries:* $\mathcal{A}$ submits the authenticated evidence δ along with a warrant m_w to $\mathcal{B}$. If δ is valid, $\mathcal{B}$ returns the converted ordinary evidence (m, Ω), else, an error symbol ¶ is outputted as a result.

Challenge: The adversary $\mathcal{A}$ produces ordinary evidence, m_0 and m_1, of the same length. The challenger $\mathcal{B}$ flips a coin $\lambda \longleftarrow \{0,1\}$ and generates authenticated evidence δ* for m_λ. The authenticated evidence δ* is then delivered to $\mathcal{A}$ as a target challenge.

Phase 2: The adversary $\mathcal{A}$ can issue new queries as those in Phase 1 except the AEV query for the target challenge.

Guess: At the end of the game, $\mathcal{A}$ outputs a bit λ'. The adversary $\mathcal{A}$ wins this game if $\lambda' = \lambda$. We define $\mathcal{A}$'s advantage as $Adv(\mathcal{A}) = |Pr[\lambda' = \lambda] - 1/2|$.

Definition 2. (Unforgeability) *A cryptographic scheme is said to achieve the security requirement of unforgeability against existential forgery under adaptive chosen-message attacks (EF-CMA) if there is no probabilistic polynomial-time adversary $\mathcal{A}$ with non-negligible advantage in the following game played with a challenger $\mathcal{B}$:*

Setup: $\mathcal{B}$ first runs the Setup(1^k) algorithm and sends the system's public parameters $params$ to the adversary $\mathcal{A}$.

Phase 1: The adversary $\mathcal{A}$ adaptively makes PCG and AEG queries as those in Phase 1 of Definition 1.

Forgery: Finally, $\mathcal{A}$ produces the authenticated evidence δ* which is not outputted by the AEG query. The adversary $\mathcal{A}$ wins if δ* is valid.

4.2. Security proof

We prove that the proposed scheme achieves the IND-CCA2 and the EF-CMA security in the random oracle model as Theorems 1 and 2, respectively. The security proofs can also be applied to its variant with message linkages, since they have almost the same structure.

Theorem 1. (Proof of Confidentiality) *The proposed scheme is (τ, q_{h_1}, q_{h_2}, q_{h_4}, q_{PCG}, q_{AEG}, q_{AEV}, ϵ)-secure against indistinguishability under adaptive chosen-ciphertext attacks (IND-CCA2) in the random oracle model if there is*

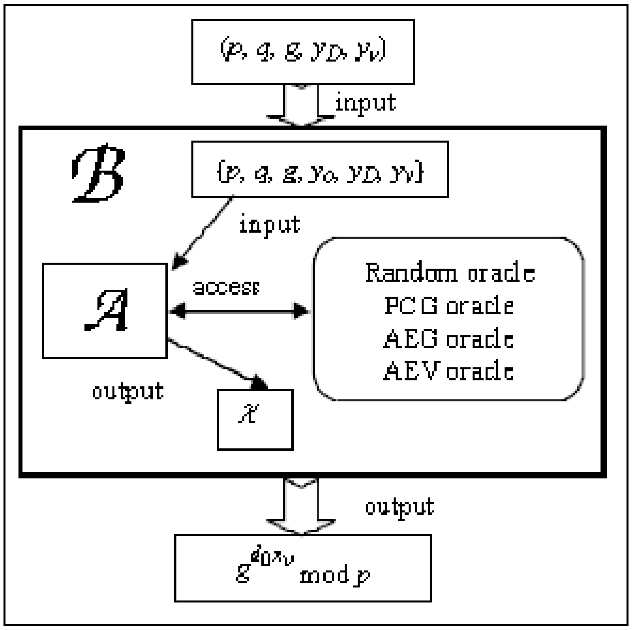

Figure 2 The proof structure of confidentiality in Theorem 1.

no probabilistic polynomial-time adversary that can (τ', ϵ')*-break the CDHP, where*

$$\epsilon' \geq q_{h_4}(q_{h_3} + q_{h_4})^{-1}(2\epsilon - q_{AEV}(q_{h_2} + q_{h_4} + 1)(2^{-k})),$$
$$\tau' \approx t_\lambda(2q_{PCG} + 4q_{AEG} + 3q_{AEV} + 4).$$

Here t_λ *is the time for performing a modular exponentiation over a finite field.*

Proof: Suppose that a probabilistic polynomial-time adversary $\mathcal{A}$ can break the proposed scheme with non-negligible advantage ϵ under CCA2 after running in time at most τ and asking at most q_{h_i} h_i random oracle (for i = 1 to 4), q_{PCG} PCG, q_{AEG} AEG and q_{AEV} AEV queries. We say that $\mathcal{A}$ (τ, q_{h_1}, q_{h_2}, q_{h_3}, q_{h_4}, q_{PCG}, q_{AEG}, q_{AEV}, ϵ)-breaks the proposed scheme under CCA2. Then we can construct another algorithm $\mathcal{B}$ that (τ', ϵ')-breaks the CDHP by taking $\mathcal{A}$ as a subroutine. Let all involved parties and parameters be defined the same as those in Section 3.1. The objective of $\mathcal{B}$ is to obtain ($g^{d_o x_v}$ mod p) by taking (p, q, g, y_D, y_v) as inputs. Fig. 2 depicts the above proof structure of this Theorem. In this proof, $\mathcal{B}$ simulates a challenger to $\mathcal{A}$ in the following game.

Setup: The challenger $\mathcal{B}$ runs the Setup(1^k) algorithm and sends the system's public parameters $params$ = p, q, g and (y_o, y_D, y_v) to the adversary $\mathcal{A}$.

Phase 1: $\mathcal{A}$ issues the following queries adaptively:

- h_1 *oracle:* When $\mathcal{A}$ makes an h_1 oracle of (m_w, T), $\mathcal{B}$ first searches the h_1-list for a matched entry; else, he chooses $v_1 \in_R Z_q$, adds (m_w, T, v_1) into the h_1-list, and then returns v_1 as a result. We use the algorithm of O-Sim_h_1(m_w, T) to express $\mathcal{B}$'s operation. The simulated random oracle O-Sim_h_1 is detailed in Fig. 3. Note that the function insert (N, b) will insert the value b into the list N.

```
oracle O-Sim_h1(m_w, T)
for i = 0 to q_h1 − 1
    if (h1_list[i] = (m_w, T, v1)) then
        exit for;
    else if (h1_list[i] = null) then
        Choose v1 ∈R Zq;
        insert(h1_list, (m_w, T, v1)); exit for;
    end if
next i
return v1;
```

Figure 3 Algorithm of the simulated random oracle O-Sim_h_1.

```
oracle O-Sim_h2(m, C, R)
for i = 0 to q_h2 − 1
    if (h2_list[i] = (m, C, R, v2)) then
        exit for;
    else if (h2_list[i] = null) then
        Choose v2 ∈R Zq;
        insert(h2_list, (m, C, R, v2)); exit for;
    end if
next i
return v2;
```

Figure 4 Algorithm of the simulated random oracle O-Sim_h_2.

- h_2 *oracle:* When $\mathcal{A}$ makes an h_2 oracle of (m, C, R), $\mathcal{B}$ first searches the h_2-list for a matched entry; else, he chooses $v_2 \in_R Z_q$, adds (m, C, R, v_2) into the h_2-list, and then returns v_2 as a result. We use the algorithm of O-Sim_h_2(m, C, R) to express $\mathcal{B}$'s operation. The simulated random oracle O-Sim_h_2 is detailed in Fig. 4.

- h_3 *oracle:* When $\mathcal{A}$ makes an h_3 oracle of $Z = CK$ mod p, $\mathcal{B}$ first searches the h_3-list for a matched entry; else, he chooses $v_3 \in_R Z_q$, adds (Z, v_3) into the h_3-list, and then returns v_3 as a result. We use the algorithm of O-Sim_h_3(Z) to express $\mathcal{B}$'s operation. The simulated ran-

```
oracle O-Sim_h3(Z)
for i = 0 to q_h3 − 1
    if (h3_list[i] = (Z, v3)) then
        exit for;
    else if (h3_list[i] = null) then
        Choose v3 ∈R Zq;
        insert(h3_list, (Z, v3)); exit for;
    end if
next i
return v3;
```

Figure 5 Algorithm of the simulated random oracle O-Sim_h_3.

```
oracle O-Sim_h4(K)
for i = 0 to q_h4 − 1
    if (h4_list[i] = (K, v4)) then
        exit for;
    else if (h4_list[i] = null) then
        Choose v4 ∈R {0, 1}^k;
        insert(h4_list, (K, v4)); exit for;
    end if
next i
return v4;
```

Figure 6 Algorithm of the simulated random oracle O-Sim_h_4.

```
oracle O-Sim_PCG(mw)
do
    Choose σ, v1 ∈R Zq;
    Compute T = g^σ yo^v1 mod p;
while (check(h1_list, (mw, T, *)) = true)
    // "*" denotes wildcard
insert(h1_list, (mw, T, v1));
    // define h1(mw, T) = v1
return (mw, σ, T);
```

Figure 7 Algorithm of the simulated PCG oracle O-Sim_PCG.

```
oracle O-Sim_AEG(m)
Choose a proper mw;
(mw, σ, T) = O-Sim_PCG(mw);
Compute C = Tg^−σ mod p; K = yv^σ mod p;
do
    Choose S, v2 ∈R Zq;
    Compute R = g^S yD^v2 mod p;
while (check(h2_list, (m, C, R, *)) = true)
insert(h2, (m, C, R, v2));
    // define h2(m, C, R) = v2
Compute W = h3(CK mod p)S mod q;
Q = O-Sim_h4(K) ⊕ m;
return {δ = (Q, W, R, T), mw};
```

Figure 8 Algorithm of the simulated AEG oracle O-Sim_AEG.

dom oracle O-Sim_h_3 is detailed in Fig. 5.

- h_4 *oracle:* When $\mathcal{A}$ makes an h_4 oracle of K, $\mathcal{B}$ first searches the h_4-list for a matched entry; else, he chooses $v_4 \in_R \{0,1\}^k$, adds (K, v_4) into the h_4-list, and then returns v_4 as a result. We use the algorithm of O-Sim_$h_4(K)$ to express $\mathcal{B}$'s operation. The simulated random oracle O-Sim_h_4 is detailed in Fig. 6.

- *PCG queries:* When $\mathcal{A}$ makes a PCG query, $\mathcal{B}$ first chooses a proper m_w and two integers $(\sigma, v_1) \in_R Z_q$ to compute $T = g^\sigma y_o^{v_1}$ mod p where $h_1(m_w, T)$ has never been queried. Then $\mathcal{B}$ adds (m_w, T, v_1) into the h_1-list, and returns (m_w, σ, T) as a result. We use the algorithm of O-Sim_$PCG(m_w)$ to express $\mathcal{B}$'s operation. The simulated PCG oracle O-Sim_PCG is detailed in Fig. 7. Note that the function **check**(N, b) will return a Boolean value depending on whether the value b is stored in the list N.

- *AEG queries:* When $\mathcal{A}$ makes an AEG query for the ordinary evidence m, $\mathcal{B}$ first obtains (m_w, σ, T) by making a PCG query and computes $C = Tg^{-\sigma}$ mod p and $K = y_v^\sigma$ mod p. Then he chooses S, $v_2 \in_R Z_q$ to compute $R = g^S y_D^{v_2}$ mod p where $h_2(m, C, R)$ has never been queried, adds (m, C, R, v_2) into the h_2-list, and derives (W, Q) as Eqs. (13) and (14), respectively. Finally, $\mathcal{B}$ returns $\delta = (Q, W, R, T)$ along with m_w as the result. We use the algorithm of O-Sim_AEG(m) to express $\mathcal{B}$'s operation. The simulated AEG oracle O-Sim_AEG is detailed in Fig. 8.

- *AEV queries:* When $\mathcal{A}$ makes an AEV query for the authenticated evidence δ with a warrant m_w, $\mathcal{B}$ first obtains v_1 by making an $h_1(m_w, T)$ query, finds out all (K_i, v_{4i})'s from the h_4-list and computes $S_i = h_3(CK_i \bmod p)^{-1}W$ mod q, for $i = 0$ to $q_{h_4} - 1$. If $h_2(*, C, R)$ has ever been queried, $\mathcal{B}$ retrieves all possible (m_j, v_{2j})'s from the h_2-list and checks if $m_j = Q \oplus v_{4i}$ and $R = g^{S_i} y_D^{v_{2j}}$ mod p. If they holds, $\mathcal{B}$ returns $\{m_j, \Omega = (S_i, R, T), m_w\}$ as a result. Otherwise, an error symbol ¶ is returned. We use the algorithm of O-Sim_$AEV(\delta, m_w)$ to express $\mathcal{B}$'s

```
oracle O-Sim_AEV(δ, m_w) // δ = (Q, W, R, T)
  v1 = O-Sim_h1(m_w, T);
  Compute C = y_o^v1 mod p;
  Find out all (K_i, v4_i)'s from the h4_list;
  Compute S_i = O-Sim_h3(CK_i mod p)^-1 W mod q;
  if (check(h2_list, (*, C, R, *)) = true) then
  // h2(*, C, R) has ever been queried
     for j = 0 to q_h2 - 1
          if (h2_list[j] = (*, C, R, *)) then
               Retrieve m_j, and v2_j
               if (m_j = Q ⊕ v4_i) and
                   (R = g^S_i y_D^v2_j mod p)) then
                   return (m_j, Ω = (S_i, R, T), m_w);
               end if
          end if
     next j
  else // h2(*, C, R) has never been queried
     return ¶
  end if
```

Figure 9 Algorithm of the simulated AEV oracle O-Sim_AEV.

```
algorithm Sim_Challenge(m_λ)
  Choose a proper m_w*;
  Choose S*, σ, v1, v2, v3 ∈_R Z_q, v4 ∈_R {0, 1}^k;
  Compute C = y_o^v1 mod p; T* = (y_D^σ)C mod p;
  insert(h1_list, (m_w, T*, v1));
    // define h1(m_w, T*) = v1
  Compute R* = g^S* y_D^v2 mod p;
  insert(h2_list, (m_λ, C, R*, v2));
    // define h2(m_λ, C, R*) = v2
  Q* = v4 ⊕ m_λ;
    //implicitly define h4(K*) = v4 where
      K* = (y_D^σ)^x_v mod p and B does not know it.
  Compute W* = v3 S* mod q;
    // implicitly define h3(CK*) = v3
  return (δ* = (Q*, W*, R*, T*), m_w*);
```

Figure 10 Algorithm of the simulated Sim_Challenge.

operation. The simulated AEV oracle O-Sim_AEV is detailed in Fig. 9.

Challenge: $\mathcal{A}$ generates ordinary evidence, m_0 and m_1, of the same length. The challenger $\mathcal{B}$ flips a coin $\lambda \longleftarrow \{0,1\}$ and chooses a proper warrant m_w*. He further chooses S*, σ, v_1, v_2, $v_3 \in_R Z_q$ and $v_4 \in_R \{0,1\}^k$ to compute $C = y_o^{v_1}$ mod p, T* = $(y_D^{\sigma})C$ mod p, R* = $g^{S^*} y_D^{v_2}$ mod p, Q* = $v_4 \oplus m_\lambda$ and W* = $v_3 S$* mod q. To ensure the consistency of simulated random oracles, $\mathcal{B}$ adds two entries (m_w, T*, v_1) and (m_λ, C, R*, v_2) into the h_1-list and h_2-list, respectively. Note that $\mathcal{B}$ has implicitly defined $h_3(CK*) = v_3$ and $h_4(K*) = v_4$, where $K* = (y_D^{\sigma})^{x_v}$ mod p and he does not know it. Finally, the authenticated evidence δ* = (Q*, W*, R*, T*) and the warrant m_w* is given to $\mathcal{A}$ as a target challenge. We use the algorithm of Sim_Challenge(m_λ) to express $\mathcal{B}$'s operation. The simulated Sim_Challenge is detailed in Fig. 10.

Phase 2: $\mathcal{A}$ makes new queries as those stated in Phase 1 except the AEV query for the target challenge δ*.

Analysis of the game: Consider the above simulations of PCG and AEG queries. One can see that simulated results are computationally indistinguishable from those generated by a real scheme. We refer the simulations of PCG and AEG queries to be perfect. Then we evaluate the simulation of AEV queries. From the algorithms of O-Sim_AEV, we find out that it is possible for an AEV query of some valid δ to return the error symbol ¶ on condition that $\mathcal{A}$ has the ability to produce δ without asking the corresponding $h_2(m_\lambda, C, R)$ or $h_4(K)$ random oracles in advance. Let AEV_ERR be the event that an AEV query returns the error symbol ¶ for some valid δ during the entire game, and VLD an event that the authenticated evidence δ submitted by $\mathcal{A}$ is valid. QH_2 and QH_4 separately denote the events that $\mathcal{A}$ has ever asked the corresponding h_2 and h_4 random oracles beforehand. Then we can express the error probability of any AEV query as

$$\begin{aligned}
&Pr[VLD|\neg QH_4 \vee \neg QH_2] \\
&= Pr[VLD|\neg QH_4] + Pr[VLD \wedge QH_4|\neg QH_2] \\
&= Pr[VLD \wedge QH_2|\neg QH_4] + Pr[VLD \wedge \neg QH_2|\neg QH_4] \\
&\quad + Pr[VLD \wedge QH_4|\neg QH_2] \\
&\leq q_{h_2}(2^{-k}) + (2^{-k}) + q_{h_4}(2^{-k}) \\
&= (q_{h_2} + q_{h_4} + 1)(2^{-k}).
\end{aligned}$$

Since $\mathcal{A}$ can make at most q_{AEV} AEV queries, we can further express the probability of AEV_ERR as

$$Pr[AEV\_ERR] \leq q_{AEV}(q_{h_2} + q_{h_4} + 1)(2^{-k}). \quad (19)$$

Additionally, in the challenge phase, $\mathcal{B}$ has returned a simulated δ* = (Q*, W*, R*, T*) where T* = $y_D^{\sigma}C$ mod p, which implies the secret K* is implicitly defined as $(y_D^{\sigma})^{x_v}$ mod p. Let GP be the event that the entire simulation game does not abort. Obviously, if the adversary $\mathcal{A}$ never asks $h_3(CK*)$ or $h_4(K*)$ random oracles in Phase 2, the entire simulation game could be normally terminated. We denote the two events that $\mathcal{A}$ does make an $h_3(CK*)$ and $h_4(K*)$ query in Phase 2 by QH_3* and QH_4*. When the entire simulation game does not abort, it can be seen $\mathcal{A}$ gains no advantage in guessing λ due to the randomness of the output of random oracles, i.e.,

$$Pr[\lambda' = \lambda|\text{GP}] = 1/2. \quad (20)$$

Rewriting the expression of $Pr[\lambda' = \lambda]$, we have

$$\begin{aligned} Pr[\lambda' = \lambda] &= Pr[\lambda' = \lambda|\text{GP}]Pr[\text{GP}] \\ &\quad + Pr[\lambda' = \lambda|\neg\text{GP}]Pr[\neg\text{GP}] \\ &\leq (1/2)Pr[\text{GP}] + Pr[\neg\text{GP}] \quad \text{(by Eq. (20))} \\ &= (1/2)(1 - Pr[\neg\text{GP}]) + Pr[\neg\text{GP}] \\ &= (1/2) + (1/2)Pr[\neg\text{GP}]. \quad (21) \end{aligned}$$

On the other hand, we can also derive that

$$\begin{aligned} Pr[\lambda' = \lambda] &\geq Pr[\lambda' = \lambda|\text{GP}]Pr[\text{GP}] \\ &= (1/2)(1 - Pr[\neg\text{GP}]) \\ &= (1/2) - (1/2)Pr[\neg\text{GP}]. \quad (22) \end{aligned}$$

With inequalities (21) and (22), we know that

$$|Pr[\lambda' = \lambda] - 1/2| \leq (1/2)Pr[\neg\text{GP}]. \quad (23)$$

Recall that in Definition 1, $\mathcal{A}$'s advantage is defined as $Adv(\mathcal{A}) = |Pr[\lambda' = \lambda] - 1/2|$. By the initial assumption, $\mathcal{A}$ has non-negligible probability ϵ to break the proposed scheme. We therefore have $\epsilon = |Pr[\lambda' = \lambda] - 1/2|$. Combining Eq. (23), we can further derive $\epsilon \leq (1/2)Pr[\neg\text{GP}]$. From the above analyses, we have known that the entire simulation game aborts, denoted as $\neg$GP, if one of the events QH_3, QH_4 and AEV_ERR occurs. Consequently, we obtain

$$\begin{aligned} \epsilon &= (1/2)(Pr[\text{QH}_3^* \vee \text{QH}_4^* \vee \text{AEV\_ERR}]) \\ &\leq (1/2)(Pr[\text{QH}_3^*] + Pr[\text{QH}_4^*] + Pr[\text{AEV\_ERR}]) \end{aligned}$$

Combining Eq. (19) and rewriting the above inequality, we get

$$\begin{aligned} (Pr[\text{QH}_3^*] + Pr[\text{QH}_4^*]) &\geq 2\epsilon - Pr[\text{AEV\_ERR}] \\ &\geq 2\epsilon - q_{AEV}(q_{h_2} + q_{h_4} + 1)(2^{-k}). \end{aligned}$$

If the event $(\text{QH}_3^* \vee \text{QH}_4^*)$ happens, we claim that the value $K* = (y_D^{\sigma})^{x_v}$ mod p will be stored in some entry of the h_4-list with the probability of $q_{h_4}(q_{h_3}+q_{h_4})^{-1}$. Consequently, $\mathcal{B}$ has non-negligible probability

$$\epsilon' \geq q_{h_4}(q_{h_3} + q_{h_4})^{-1}(2\epsilon - q_{AEV}(q_{h_2} + q_{h_4} + 1)(2^{-k}))$$

to output $K^{*\sigma^{-1}} = g^{d_0 x_v}$ and solve the CDHP. The computational time required for $\mathcal{B}$ is $\tau' \approx \tau + t_\lambda(2q_{PCG} + 4q_{AEG} + 3q_{AEV} + 4)$.

Q.E.D.

In 2000, Pointcheval and Stern introduced the Forking lemma [21] to prove the security for generic digital signature schemes in the random oracle model. If we apply their techniques to prove our scheme, we can first obtain two equations below:

$$R = g^S y_D^{h_2(m,C,R)} \bmod p,$$
$$R = g^{S'} y_D^{h_2'(m,C,R)} \bmod p.$$

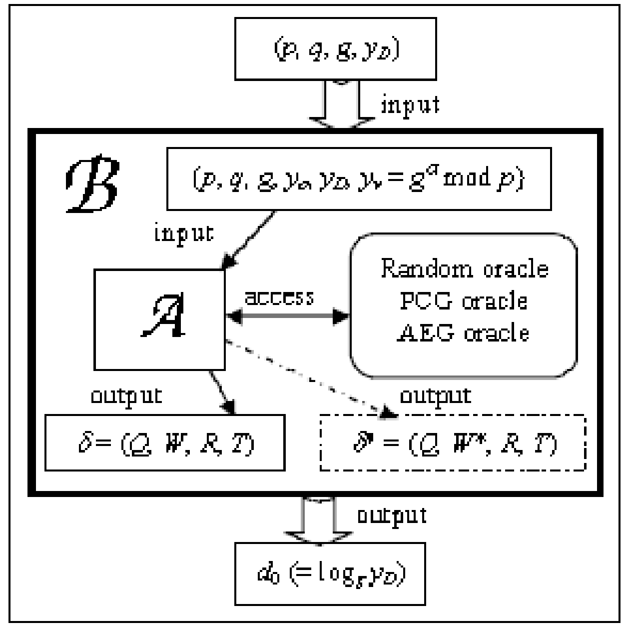

Figure 11 The proof structure of unforgeability in Theorem 2.

By combining the above two equalities, we can further derive the private key d_0 as

$$d_0 = (S - S')/(h_2'(m, C, R) - h_2(m, C, R))$$

and hence solve the DLP for the instance $(p, q, g, y_D = g^{d_0}$ mod $p)$.

Still, to give a tight reduction from the hardness of DLP to our proposed scheme, we present another more detailed security proof and the advantage analysis as Theorem 2.

Theorem 2. (Proof of Unforgeability) *The proposed scheme is (τ, q_{h_1}, q_{h_2}, q_{PCG}, q_{AEG}, ϵ)-secure against existential forgery under adaptive chosen-message attacks (EF-CMA) in the random oracle model if there is no probabilistic polynomial-time adversary that can (τ', ϵ')-break the DLP, where*

$$\epsilon' \geq 4^{-1}(\epsilon - 2^{-2k})^3(q_{h_2}^{-1}),$$
$$\tau' \approx \tau + t_\lambda(4q_{PCG} + 10q_{AEG}).$$

Here t_λ is the time for performing a modular exponentiation over a finite field.

Proof: Suppose that a probabilistic polynomial-time adversary $\mathcal{A}$ breaks the proposed scheme with non-negligible advantage ϵ under CMA after running in time at most τ and asking at most q_{h_i} h_i random oracle (for i = 1 to 4), q_{PCG} PCG and q_{AEG} AEG queries. We say that $\mathcal{A}$ (τ, q_{h_1}, q_{h_2}, q_{h_3}, q_{h_4}, q_{PCG}, q_{AEG}, ϵ)-breaks the proposed scheme under CMA. Then we can construct another algorithm $\mathcal{B}$ that (τ', ϵ')-breaks the DLP by taking $\mathcal{A}$ as a subroutine. Let all involved parties and notations be defined the same as those in Section 4.1. The objective of $\mathcal{B}$ is to obtain $d_0(=\log_g y_D)$ by taking (p, q, g, y_D) as inputs. Fig. 11 depicts the above proof structure of this Theorem. In this proof, $\mathcal{B}$ simulates a challenger to $\mathcal{A}$ in the following game.

Setup: The challenger $\mathcal{B}$ runs the Setup(1^k) algorithm to obtain the system's public parameters $params$ = $\{p, q, g\}$ and comes up with a random tape composed of a long sequence of random bits. Then $\mathcal{B}$ simulates two runs of the proposed scheme to the adversary $\mathcal{A}$ on input $params$, y_o, y_D, y_v = g^α mod p where $\alpha \in_R Z_q$, and the random tape.

Phase 1: $\mathcal{A}$ adaptively asks h_i, for i = 1 to 4, random oracle, PCG and AEG queries as those defined in Theorem 1.

Analysis of the game: According to the analyses of Theorem 1, the simulations of PCG and AEG queries are perfect. Namely, the adversary $\mathcal{A}$ cannot distinguish whether he is playing in either a simulation or a real scheme. Let VLD be the event that $\mathcal{A}$ forges a valid authenticated evidence δ = (Q, W, R, T) along with a warrant m_w for his arbitrarily chosen m. Since $\mathcal{A}$ has non-negligible probability ϵ to break the proposed scheme under CMA by the initial assumption, we know that

$$Pr[\text{VLD}] = \epsilon.$$

Now we further consider the situation where $\mathcal{A}$ is able to output a valid δ without asking h_2 random oracles in advance. Let $(\neg\text{QH}_2)$ be the event that $\mathcal{A}$ guesses correct output value of $h_2(m, C, R)$ without asking the random oracle, i.e., $Pr[\neg\text{QH}_2] \leq 2^{-k}$. Then, we can express the probability that $\mathcal{A}$ outputs a valid forgery δ = (Q, W, R, T) after asking h_2 random oracle as

$$Pr[\text{VLD} \wedge \text{QH}_2] \geq (\epsilon - 2^{-k}).$$

With the initially selected private key α, $\mathcal{B}$ can recover m and obtain the parameter S.

Then $\mathcal{B}$ launches the second simulation. He again runs $\mathcal{A}$ on the same input. Since the adversary $\mathcal{A}$ is given the same sequence of random bits, we can anticipate that the i-th random query $\mathcal{A}$ asks will always be the same as the one in the first simulation. In the second simulation, $\mathcal{B}$ returns identical results as those he responds in the first time until $\mathcal{A}$ makes the $h_2(m, C, R)$ query. At this time, $\mathcal{B}$ directly gives another answer $v_2^* \in_R Z_q$ rather than original v_2. Meanwhile, $\mathcal{A}$ is then supplied with a different random tape which also consists of a long sequence of random bits. From the statement of "Forking lemma", we can learn that when $\mathcal{A}$ finally makes another valid forgery δ^* = (Q, W^*, R, T) where $h_2(m, C, R) \neq h_2^*(m, C, R)$, $\mathcal{B}$ could solve the DLP with non-negligible probability. To analyze $\mathcal{B}$'s success probability, we use the "Splitting lemma" [21] described below:

Let X and Y be the sets of possible sequences of random bits and random function values provided to $\mathcal{A}$ before and after the $h_2(m, C, R)$ query is issued, respectively. It follows that on inputting a random value $(x \parallel y)$ for any $x \in X$ and $y \in Y$, $\mathcal{A}$ returns a valid forgery with non-negligible probability ϵ, i.e.,

$$Pr_{x\in X, y\in Y}[\text{VLD}] = \epsilon.$$

By the "Splitting lemma", there exists a subset $D \in X$ such that

(a). $Pr[x \in D] = |D| \cdot |X|^{-1} \geq 2^{-1}\epsilon$.
(b). $\forall x \in D, Pr_{y\in Y}[\text{VLD}] \geq 2^{-1}\epsilon$.

If we let $\rho \in D$ and $y' \in Y$ separately be the supplied sequences of random bits and random function values before and after $\mathcal{A}$ makes the $h_2(m, C, R)$ query, $\mathcal{A}$ is able to make a valid forgery in the second simulation with the probability of at least $(2^{-1}\epsilon)^2$ = $4^{-1}\epsilon^2$, i.e.,

$$Pr_{\rho\in D, y'\in Y}[\text{VLD}] \geq 4^{-1}\epsilon^2.$$

Since we have known that $\mathcal{A}$ eventually returns another valid δ^* = (Q, W^*, R, T) with $h_2(m, C, R) \neq h_2^*(m, C, R)$ is $q_{h_2}^{-1}$, the probability of $\mathcal{B}$ to solve the DLP in the second simulation can be represented as

$$\begin{aligned}\epsilon' &\geq (\epsilon - 2^{-k})(4^{-1}(\epsilon - 2^{-k})^2)(q_{h_2}^{-1}) \\ &= (4q_{h_2})^{-1}(\epsilon - 2^{-k})^3.\end{aligned}$$

Moreover, the computational time required for $\mathcal{B}$ in one simulation is

$$\tau' \approx \tau + 2t_\lambda(2q_{PCG} + 4q_{AEG}).$$

Q.E.D.

According to Theorem 2, the proposed scheme is secure against existential forgery attacks. That is, the private key cannot be forged and the group for judicial policemen cannot repudiate generated authenticated evidence. Hence, we obtain the following corollary.

Corollary 1. *The proposed scheme satisfies the security requirement of non-repudiation.*

5. Conclusion

From the perspective of realistic consideration, we proposed an efficient and secure data protection method for computer forensics in this paper. Our design idea is motivated by the practical forensic procedure in legal cases. To provide better flexibility, the proposed scheme equips any t-out-of-n judicial policemen to cooperatively generate valid authenticated evidence on behalf of the investigating authority rather than the whole group. For facilitating the encryption/verification of large evidence, a variant with message linkages is also introduced by dividing it into many smaller blocks. Compared with existing related cryptographic mechanisms that take both the properties of confidentiality and authenticity into consideration, our method provides better functionalities and efficiency. Moreover, to guarantee the feasibility of proposed work, we also proved that the proposed scheme achieves the IND-CCA2 and the EF-CMA security in the random oracle model.

References

[1] S. Araki, S. Uehara and K. Imamura, The limited verifier signature and its application, *IEICE Transactions on Fundamentals*, **E82-A**, 1999, pp. 63-68.
[2] F. Cao and Z. Cao, Cryptanalysis on a proxy multi-signature scheme, *Proceedings of the 1st International Multi-Symposiums on Computer and Computational Sciences (IMSCCS'06)*, **2**, IEEE Press, Piscataway, USA, 2006, pp. 117-120.
[3] T. Y. Chang, A convertible multi-authenticated encryption scheme for group communications, *Information Sciences*, **178**, 2008, pp. 3426-3434.
[4] H. Delfs and H. Knebl, *Introduction to Cryptography: Principles and Applications*, Springer, 2002.
[5] W. Diffie and M. Hellman, New directions in cryptography, *IEEE Transactions on Information Theory*, **IT-22**, 1976, pp. 644-654.
[6] T. ElGamal, A public key cryptosystem and a signature scheme based on discrete logarithms, *IEEE Transactions on Information Theory*, **IT-31**, 1985, pp. 469-472.
[7] P. Horster, M. Michel and H. Peterson, Authenticated encryption schemes with low communication costs, *Electronics letters*, **30**, 1994, pp. 1212-1213.
[8] F. Hou, Z. Wang, Y. Tang and Z. Liu, Protecting integrity and confidentiality for data communication, *Proceedings of 9th International Symposium on Computers and Communications (ISCC)*, **1**, 2004, pp. 357-362.
[9] S. J. Hwang and C. C. Chen, A new multi-proxy multisignature scheme, *2001 National Computer Symposium*, 2001, pp. 19-26.
[10] S. J. Hwang and C. H. Shi, A simple multi-proxy signature scheme, *Proceedings of the 10th National Conference on Information Security*, 2000, pp. 134-138.
[11] W. H. Holzmann and H. Kharaghani, Weak amicable T-matrices and Plotkin arrays, *Journal of Combinatorial Designs*, **16**, 2008, pp. 4452.
[12] ML Das, A Saxena, VP Gulati, A dynamic ID-based remote user authentication scheme, *Consumer Electronics, IEEE Transactions on* , **50**, 2004, pp. 629-631.
[13] C. Y. Lin, T. C. Wu and J. J. Hwang, Multi-proxy signature schemes for partial delegation with cheater identification, *Proceedings of the 2nd International Workshop for Asia Public Key Infrastructure (IWAP 2002), Technical Session E: Mobility & Certification*, IOS Press, Netherlands, 2002.
[14] J. Lv, X. Wang and K. Kim, Practical convertible authenticated encryption schemes using self-certified public keys, *Applied Mathematics and Computation*, **169**, 2005, pp. 1285-1297.
[15] M. Mambo, K. Usuda and E. Okamoto, Proxy signature for delegating signature operation, *Proceedings of the 3rd ACM Conference on Computer and Communications Security*, ACM press, 1996, pp. 48-57.
[16] M. Mambo, K. Usuda and E. Okamoto, Proxy signatures: delegation of the power to sign messages, *IEICE Transactions on Fundamentals of Electronic Communications and Computer Science*, **E79-A**, 1996, pp. 1338-1354.
[17] W. Mao, *Modern Cryptography: Theory and Practice*, Prentice Hall, 2004.
[18] B. Meng, S. Wang, Q. Xiong, A fair non-repudiation protocol, *The 7th International Conference on Computer Supported Cooperative Work in Design*, 2002, pp. 68-73.
[19] HY Chien, CH Chen, Mutual authentication protocol for RFID conforming to EPC Class 1 Generation 2 standards, *Computer Standards & Interfaces*, *Elsevier*, **29**, 2007, pp. 254-259.
[20] Y. Q. Peng, S. Y. Xie, Y. F. Chen, R. Deng and L. X. Peng, A publicly verifiable authenticated encryption scheme with message linkages, *Proceedings of the 3rd International Conference on Networking and Mobile Computing, ICCNMC*, Zhangjiajie, China, 2005, pp. 1271-1276.
[21] D. Pointcheval and J. Stern, Security arguments for digital signatures and blind signatures, *Journal of Cryptology*, **13**, 2000, pp. 361-369.
[22] R. Rivest, A. Shamir and L. Adleman, A method for obtaining digital signatures and public-key cryptosystems, *Communications of the ACM*, **21**, 1978, pp. 120-126.
[23] A. Shamir, How to share a secret, *Communications of the ACM*, **22**, 1979, pp. 612-613.
[24] W. Stallings, *Cryptography and Network Security: Principles and Practices*, 4th Ed., Pearson, 2005.
[25] J. L. Tsai, Convertible multi-authenticated encryption scheme with one-way hash function, *Computer Communications*, **32**, 2009, pp. 783-786.
[26] S. F. Tzeng, C. Y. Yang and M. S. Hwang, A nonrepudiable threshold multi-proxy multi-signature scheme with shared verification, *Future Generation Computer Systems*, **20**, 2004, pp. 887-893.
[27] B. Wendroff, *Theoretical Numerical Analysis*, Academic Press Inc., 1996.
[28] T. S. Wu and C. L. Hsu, Convertible authenticated encryption scheme, *The Journal of Systems and Software*, **62**, 2002, pp. 205-209.
[29] T. S. Wu, C. L. Hsu, K. Y. Tsai, H. Y. Lin and T. C. Wu, Convertible multi-authenticated encryption scheme, *Information Sciences*, **178**, 2008, pp. 256-263.
[30] T. S. Wu and H. Y. Lin, A group-oriented proxy CMAE scheme with computational secrecy, *International Journal of Innovative Computing, Information and Control*, **4**, 2008, pp. 3037-3047.
[31] Q. Xue and Z. Cao, A nonrepudiable multi-proxy multisignature scheme, *Proceedings of 1st Joint Workshop on Mobile Future & Symposium on Trends in Communications (SympoTIC '04)*, IEEE Press, Piscataway, USA, 2004, pp. 102-105.
[32] Q. Xue and Z. Cao, Improvement of multi-proxy signature scheme, *Proceedings of the 4th International Conference on Computer and Information Technology (CIT'04)*, IEEE Press, Piscataway, USA, 2004, pp. 450-455.
[33] A. P. DAWID, J. MORTERA, V. L. PASCALI and D. VAN BOXEL, Probabilistic Expert Systems for Forensic Inference from Genetic Markers, *Scandinavian Journal of Statistics*, **29**, 2002, pp. 577-595.
[34] L. B. Yi and G. Xiao, Proxy multisignature scheme: a new type of proxy signature scheme, *Electronics Letters*, **36**, 2000, pp. 527-528.
[35] Y. Zheng, Digital signcryption or how to achieve cost(signature & encryption) $\ll$ cost(signature) + cost(encryption), *Advances in Cryptology - CRYPTO'97*, Springer-Verlag, 1997, pp. 165-179.

Tzong-Sun Wu received his BS degree in electrical engineering from the National Taiwan University, Taiwan in 1990, and his PhD in information management from the National Taiwan University of Science and Technology, Taiwan in 1998. From August 1998 to July 2001, he has been an Assistant Professor in the Department of Information Management of Huafan University. From August 2001 to January 2007, he has been an Associate Professor in the Department of Informatics of Fo Guang University. He is now with the Department of Computer Science, National Taiwan Ocean University. His research interests include information security, watermarking, digital right management, and e-commerce.

Han-Yu Lin received BA degree in economics from the Fu-Jen University, Taiwan in June 2001, his MS degree in information management from the Huafan University, Taiwan in June 2003, and his Ph.D. degree in computer science and engineering from the National Chiao Tung University, Taiwan in December 2010. He served as a research assistant in the Department of Computer Science and Engineering, National Taiwan Ocean University, Taiwan from March 2011 to December 2011. He was a senior engineer in CyberTrust Technology Institute, Institute for Information Industry, Taiwan from January 2012 to July 2012. Since August 2012, he has been an Assistant Professor in the Department of Computer Science and Engineering, National Taiwan Ocean University, Taiwan. His research interests include cryptology, network security, digital forensics, RFID privacy and application, cloud computing security and e-commerce security.