A Provably Secure Trapdoor Hash Function Based on *k*-ECAA

Fuw-Yi Yang[a,*], Su-Hui Chiu[b]

[a]Computer Science and Information Engineering,
Chaoyang University of Technology, Taiwan,

[b]Office of Accounting,
Chaoyang University of Technology, Taiwan,

[*]*Corresponding author's email: yangfy [AT] cyut.edu.tw*

ABSTRACT--- ***The integration of trapdoor hash function and scheme of digital signature not only enhances the security of the signature scheme, but also reduces the online computation during the construction of signatures. Many schemes of trapdoor hash function have been proposed. However, many of them are not provably secure. This paper proposes a trapdoor hash function based on an extension of k-CAA assumption, i.e. k-ECAA. On the assumption of random oracle model and adaptively chosen message attack, a forgery of collision implies solution of k-ECAA instance.***

Keywords---- Digital signature, k-CAA assumption, k-ECAA assumption, Trapdoor hash function.

1.INTRODUCTION

Hash functions are commonly applied in digital signatures. Digital signing can be divided into three major phases: devising signature keys, signing documents and generating signatures, and certifying signatures. In a typical document-signing process, hash functions are first used to extract the abstract of the document to be signed, and the abstract is then digitally signed.

Collision-resistance is a crucial property of traditional hash functions but is selectively established for chameleon functions [Krawczyk& Rabin, 2000; Chen, 2014] or **trapdoor hash functions**[Shamir & Tauman, 2001; Yang, 2009 and 2013]. This indicates that trapdoor key attackers are unknown, and no algorithms of polynomial time complexity are available for calculating collision information (collided preimages). However, trapdoor key holders can efficiently identify other collision information and generate identical hash. For example, assume $TH(\cdot)$ represents the trapdoor hash functions, and hash value $v = TH(h_1)$. After determining h_1, trapdoor key holders can calculate h_2, causing$v = TH(h_2) = TH(h_1)$.

This paper presents the design and practical exampleof a trapdoor hash functions on the basis of mathematics problem: collusion attack algorithms with k traitors (k-CAAs). In practical example, the system parameter designs, trapdoor hash functions, and security properties are discussed.

2.INTRODUCTION TOCOLLUSION ATTACK ALGORITHMS WITH K TRAITORS

The pairing function e is the primary element inbilinear pairing cryptosystems and pairs the elements in groups G_1 and G_2 to another group G_T, specifically e: $G_1 \times G_2 \rightarrow G_T$.$G_1$ and G_2 are typically expressed as additive groups, and G_T is expressed as a multiplicative group. The three groups consist of identical orders, which are the large prime q. In practice, the EC additive group$G_1 = G_2$ can typically be adopted. G_T is the multiplicative group of a finite field. The pairing function e adopts either Weil or Tate pairing and comprises bilinear and non degenerative properties:

Bilinearity: Bilinearity is found among the elements P, P_1, and $P_2$of the additive group G_1 and Q, Q_1, and Q_2 of group G_2.

$$e(P_1 + P_2, Q) = e(P_1, Q) \cdot e(P_2, Q)$$
$$e(P, Q_1 + Q_2) = e(P, Q_1) \cdot e(P, Q_2)$$

Non degenerative property: For themultiplicative groupG_T, 1 is used as the unit element, whereas O denotes the unit element used in the additive groups G_1 and G_2. For all elements P in G_1, $e(P, Q) = 1$; otherwise,$Q = O$. For all elements Q in G_2, $e(P, Q) = 1$;otherwise,$P = O$.

When $G_1 = G_2$ and $P \in G_1$, $e(P, P)$ generates the multiplicative groupG_T. For $Q \in G_1$, $c = e(P, Q) \in G_T$ can be calculated. However, provided $c \in G_T$ and $Q \in G_1$, calculating $P \in G_1$, causes$c = e(P, Q)$to be difficult.

Bilinear pairing enables the DLP of G_1 to be simplified to that of G_T, suggesting that the DLP for$G_1$is not more difficult than that for $G_T$is.Moreover, G_1 demonstrates that decisional Diffie–Hellman problems [Boneh, 1998] can be easily solved, whereas computational Diffie–Hellman problems remain challenging, which is referred to as gap groups [Okamoto & Pointcheval, 2001]. These results facilitate the security certification of crytographic protocols. Researchers have recently studied bilinear pairing cryptosystems [Boneh, Lynn, &Shacham, 2001; Lin, 2010; Hoffstein, Pipher, & Silverman, 2014]. These studies have revealed crucial applications of these cryptosystems, including the *k*-CAA[Mitsunari, Sakai,&Kasahara, 2002; Yang, Ma,& Wang, 2006; Dutta, Barua, & Sarkar, 2004;Tso, Yi, & Hung, 2008; Tsai, Wuand, & Hsu, 2011].

Definition 2.1.*k*-CAA problems: k is an integer, $P \in G_1$, and $x \in_R Z_q$.$\{P, Q = xP, h_1, h_2, \ldots, h_k \in Z_q, P/(h_1 + x), P/(h_2 + x), \ldots, P/(h_k + x)\}$ is provided, and $P/(h + x)$ is calculated, in which $h \notin \{h_1, h_2, \ldots, h_k\}$.

The symbol(ε, t)-break-k-CAA denotes the presence of an algorithm A that solves the k-CAA problem within the time limit t with a probability no less than ε.

Definition 2.2.(ε, t)-*k*-CAA assumption: No (ε, t)-break-k-CAA algorithms exist in cyclic group G_1.

The numerator of $P/(h_i + x)$ is multiplied by (r_i/h_i), in which $r_i \in_R Z_q$. The originally defined k-CAA problem is rewritten. K is an integer, $P \in G_1$, $x \in_R Z_q$, and $\{P, P_{Pub} = x \cdot P, R_{2k} = \{(h_i, r_i)\} | (h_i, r_i) \in \{Z_q^*\}^2$, and $i = 1, 2, \ldots, k\}$ as well as $\{P/(r_1/h_1 \cdot x + r_1), P/(r_2/h_2 \cdot x + r_2), \ldots,$ and $P/(r_k/h_k \cdot x + r_k)\}$ are provided. This outputs are $(h, r) \in \{Z_q^*\}^2$ and $P/(r/h \cdot x + r) \in G_1$, in which $(h, r) \notin R_{2k}$. Additionally, the relationship between $P/(r_i/h_i \cdot x + r_i) \in G_1$ and (h_i, r_i)can be specified using the equation $e(P/(r_i/h_i \cdot x + r_i), r_i/h_i \cdot P_{Pub} + r_i \cdot P) = e(P, P)$, thus obtaining the following modified k-CAA **problem**:

Definition 2.3. The extension of the collusion attack algorithms with *k*-traitors (*k*-ECAA) problem: k is an integer, $P \in G_1$, $x \in_R Z_q$.$\{P, P_{Pub} = x \cdot P, R_{2k} = \{(h_i, R_i) | (h_i, R_i) \in Z_q^* \times G_1$, and $i = 1, 2, \ldots, k\}$, and $S_{1k} = \{S_i | e(S_i, h_i \cdot P_{Pub} + R_i) = e(P, P)$, $(h_i, R_i) \in R_{2k}$and $i = 1, 2, \ldots, k\}$ are provided, outputting$(h, R, S) \in Z_q^* \times \{G_1\}^2$, in which$(h, R) \notin R_{2k}$.

Inference 2.4.(ε, t)-*k*-ECAA assumption: k-ECAA **problems** are not less complicated than$k$-CAA **problems**.

Proof. The relationship between $S_i = P/(r_i/h_i \cdot x + r_i) \in G_1$ and (h_i, r_i, R_i) is specified using the equation $e(P/(r_i/h_i \cdot x + r_i), r_i/h_i \cdot P_{Pub} + R_i) = e(P, P)$. A unique R_i can satisfy every h_i equation that is provided. Subsequently, the numerator of $P/(h_i + x)$ is multiplied by (r_i/h_i), in which $r_i \in_R Z_q$. Because r_i is a random number, multiplying the numerator by a random number cannot reduce the complexity of the problem. Therefore, k-ECAA **problems** are not less complicated than k-CAA **problems**.

3. TRAPDOOR HASH FUNCTION BASED ON K-ECAA

This section elucidates the k-ECAA trapdoor hash functions, which comprise algorithms for setting system parameters and keys as well as generating and verifying collision information:

Parameter Setup: Operating cyclic groups G_1 and G_T, bilinear pairing function e, and hash functions $H(\cdot)$: $\{0, 1\}^* \rightarrow Z_q$ are selected, in which $H(\cdot)$ must satisfy the properties of secure hash functions. Next, the system parameters $params = (G_1, G_2, e, q, P)$ and $H(\cdot)$ are made public.

Key setup: The user ID_u selects and secretly collects the trapdoor key $u \in_R Z_q$ and calculates and makes public the validation key$U_1 = uP \in G_1$.

Generating collision information: ID_u owns the trapdoor key and therefore can efficiently calculate the collision information (R, S) to restore the validation key. Provided that ID_u has formulated message (m), the collision information can be calculated as follows:

$r \in_R Z_q$, $R = r \cdot P \in G_1$
$h = H(m, R)$

$$S = P/(hu + r) \in G_1$$

The validation keyU_1 and collision information(R, S) are equivalent to the public signature information P_{pub} and signature[Zhang, Safavi-Naini, &Susilo, 2004], but a probability approach is adopted here to generate the collision information.

Verifying collision information: When the authenticator receives the collision information(R, S) and message m, the checking procedure is performed as follows:

$$h = H(m, R)$$
$$e(S, hU_1 + R) \stackrel{?}{=} e(P, P)$$

where, $\stackrel{?}{=}$ denotes testing the equivalence of the terms on the two sides of the symbol. When the two sides equate, authentication succeeds, and (R, S, m) is confirmed as the collision information and message of the validation keyU_1;otherwise, authentication fails. The term $e(P, P)$ on the right-hand side of the certification equation may demonstrate various concepts other than those of the restored validation keyU_1, but this difference results from the simplification process. When $S= uP/(hu + r)$ is adopted to calculate the collision information, the right-hand side of the certification equation becomes $e(U_1, P) = e(P, P)^u$, which is the restored validation key.

4. SECURITY OF THEPROPOSED TRAPDOOR HASH FUNCTION

The model assumptions of ROM [Bellare and Rogaway, 1993] and adaptively chosen message attack [Goldwasser, Micali, and Rivest, 1988]are adopted to prove the security of the proposed scheme. Theorem 4.1.proves that when the (ε, t, q_h, q_s)-forge algorithm exists in the k-CAA trapdoor hash functions, the (ε', t')-break-k-ECAA algorithm also exist in the cyclic group G_1, in which $t' = t$, and $\varepsilon' = \varepsilon (q_s/q_h)^{qs}$.

Theorem 4.1. If the k-ECAA trapdoor hash function is (ε, t, q_h, q_s)-forge, then the (ε', t')-break-k-ECAA algorithm exists in the cyclic groupG_1, in which $\varepsilon' = \varepsilon (q_s/q_h)^{qs}$, $t' = t$, and $k > q_s$.

Proof. Symbol A is used to represent the **challenger**, and F is used to represent the **forger**(ε, t, q_h, q_s)-forge. A is provided with the system parameters *params* = (G_1, G_2, e, q, P, $H(\cdot)$), an actual k-ECAA example $\{P, P_{Pub} = x \cdot P, R_{3k} = \{(h_i, m_i, R_i) | h_i = H(m_i, R_i) \in Z_q^*, m_i \in \{0, 1\}^*, R_i \in G_1$,and $i = 1, 2, \ldots, q_s\}$, and $S_{1k} = \{S_i | e(S_i, h_i \cdot P_{Pub} + R_i) = e(P, P), h_i = H(m_i, R_i)$,$(h_i, m_i, R_i) \in R_{3k}$, and $i = 1, 2, \ldots, q_s\}$. A simulates hash functions and generates hashes and signature information to respond to the demand of **Forger** F. The aim is to use the ability of F to forge to calculate (h, m, R) and S, causing $h = H(m, R)$ and $e(S, h \cdot P_{Pub} + R) = e(P, P)$, in which $(h, m, R) \notin R_{3k}$.

Public key: P denotes the public key, $P_{Pub} \in G_1$, and *params* represents the system parameters.

A prepares the hash table T_h: A generates a hashed set $H_s = \{h_1, h_2, \ldots, h_{qS}\}$, in which $H_q = \{a_i \in_R Z_q^* | i = 1, 2, \ldots, q_h\text{-}q_s\}$ and $\varnothing = H_s \cap H_q$. The hash table$T_h$ has a $q_h \times 4$ format. For $j = 1, 2, \ldots, q_h$, A arbitrarily selects elements from the union of sets H_s and H_q and places the elements into $T_h[j, 1]$ without repetition. When $T_h[j, 1] = h_i$, selections are made in set H_s; corresponding signature information S_i is then filled into $T_h[j, 2]$, and message (m_i, R_i) is filled into $T_h[j, 3]$ and $T_h[j, 4]$.

***F* requests the hashes for (m_i, R_i):** A searches the hash table$T_h[j, 3]$ according to sequence, beginning from $j = 1$. For any corresponding data on the jth row that satisfies $T_h[j, 3] = m_i$ and$T_h[j, 4] = R_i$, $T_h[j, 1]$ is returned to F; otherwise, A fills (m_i, R_i) into $T_h[j, 3]$ and $T_h[j, 4]$ and returns $T_h[j, 1]$ to F, in which $j \le q_h$ is the minimal integer rendering $T_h[j, 3]$ and $T_h[j, 4]$ as null data.

***F* requeststhe signature information for message m_i:** Assuming the hashes for (m_i, R_i) have been designated, A searches the hash table $T_h[j, 3]$ beginning from $j = 1$ to identify the corresponding data $T_h[j, 3] = m_i$ on the jth row. If$T_h[j, 2]$ is null data, A fails; otherwise, signature information $T_h[j, 2]$ and $T_h[j, 4]$ are returned to F.

Forger*F* can request the hash and signature information for arbitrary information (m_i, R_i) at any time and may ultimately generate forged signature information (m^*, R^*, S^*) for A, in which $m^* = m_{i^*}$ and $1 \le i^* \le q_h$. Through q_s number ofrequests for signature information, the condition $H(m^*, R^*) \notin H_s$ can be confirmed. Therefore, A obtains (h, m, R) and S, causing $h = H(m, R)$ and $e(S, h \cdot P_{Pub} + R) = e(P, P)$ (R, S) $P/(h + x)$, in which $h \notin H_s = \{h_1, h_2, \ldots, h_{qs}\}$. Because A generates signature information at a success rate of q_s/q_h, the probability of successful requests for signature information by F after q_s trials is $(q_s/q_h)^{qs}$. Specifically, when F can generate the (ε, t, q_h, q_s)-forge signature information, the algorithm (ε', t')-break-k-ECAA exists in cyclic groupG_1, in which $\varepsilon' = \varepsilon (q_s/q_h)^{qs}$, and $t' = t$.□

5. CONCLUSIONS

Trapdoor hash functions can aid any signature scheme to securely generate signatures online efficiently. This paper extends the assumption of *k*-CAA to the *k*-ECAA. The later assumption is at least as hard as the first assumption. Then, a trapdoor hash function based on the intractability of *k*-ECAA is proposed. The discussion of security uses the ROM and adaptively chosen message attack to simulate the adversary's attack. A forgery of hash collision means an instance of *k*-ECAA is solved. Thus, certifies the security of the proposed scheme.

6. REFERENCES

- M. Bellare and P. Rogaway, "Random oracles are practical: a paradigm for designing efficient protocols," *Proc. of the 1st ACM Conference on Computer and Communications Security CCS'93*, ACM press, pp. 62-73, 1993.
- D.Boneh, "The decisional diffie-hellman problem," *Proceedings of the Third Algorithmic NumberTheory Symposium*, LNCS 1423, pp. 48–63, 1998.
- D. Boneh, B. Lynn, and H. Shacham, "Short signatures from Weil pairing," *Advances in Cryptology-ASIACRYPT'01*, LNCS 2248, pp. 514-532, 2001.
- X. Chen, F. Zhang, W. susilo, H, Tian, J. Li, and K. Kim, "Identity-based chameleon hashing and signatures without key exposure,"*Information Sciences*,Vol. 265, pp. 198-210, 2014.
- R.Dutta, R.Barua, and P.Sarkar, "Pairing-Based Cryptographic Protocols: A Survey," available at http://eprint.iacr.org/2004/064.

- S. Goldwasser, S. Micali, and R. Rivest, "A digital signature scheme secure against adaptive chosen-message attacks," *SIAM J. Computing*, Vol. 17, No. 2, pp. 281-308, 1988.
- J.Hoffstein, J.Pipher, J. H. Silverman, "An Introduction to Mathematical Cryptography," Springer, 2014.
- H. Krawczyk and T. Rabin, "Chameleon signatures," *Symposium on Networkand Distributed Systems Security* (NDSS'00), pp.143-154, 2000.
- J. S. Lin,"橢圓曲線 Pairings 之密碼應用原理,"*Communications of the CCISA*, Vol. 16, No. 4, Oct, pp. 32-44, 2010.
- S. Mitsunari, R. Sakai and M. Kasahara, "A new traitor tracing,"*IEICE Trans. on Fundamentals*, Vol.E85-A, no.2, pp.481-484, 2002.

- T. Okamoto and D. Pointcheval, "*The gap-problems: a new class of problemsfor the security of cryptographic Schemes*," Public Key Cryptography-PKC 2001, LNCS 1992, pp. 104-118, 2001.
- Shamir and Y. Tauman, "Improved online / offline signature schemes,"*Advances in Cryptology-CRYPTO'01*, LNCS 2139, pp.355-367, 2001.
- K. Y. Tsai, T. C. Wu, and C. L. Hsu, "New secret key traitor tracing scheme with dispute settlement from bilinear maps,"*International Journal of Innovative Computing, Information and Control*, vol. 7, no.3, pp.1499-1510, 2011.
- R.Tso, X. Yi, and X. Huang, "Efficient and Short Certificateless Signature,"*Cryptology and Network Security:7th International Conference*, CANS 2008, Hong-Kong, China, December 2-4, 2008. Proceedings, LNCS 5339, pp 64-79, 2008.
- Yang, W. Ma and X. Wang, "New traitor tracing scheme against anonymous attack,"*Proc. of the1st International Conference on Innovative Computing, Information and Control*, Beijing, China,pp.389-392, 2006.
- F. Y. Yang andZ. W. Liu, "Improvement of an efficient proxy blind signature scheme,"*Fourth International Conference on InnovativeComputing, Information and Control*(ICICIC 2009), Kaohsiung, Taiwan, December 7 - 9, pp. 733-736, 2009.
- F. Y. Yang, "Improvement on a trapdoor hash function,"*International Journal of Network Security*, Vol. 9, No. 1, July, pp. 17-21, 2009.
- F. Y. Yang and L. R. Liang, "A proxy partially blind signature scheme with proxy revocation,"*Journal of Ambient Intelligence and Humanized Computing* (AIHC),Springer-Verlag,Vol. 4, Issue 2, pp. 255-263, April, 2013.
- F. Zhang , R. Safavi-naini , and W Susilo, "An efficient signature scheme from bilinear pairings and its applications," Public Key Cryptography - PKC 2004, LNCS 2947, pp. 277-290, 2004.