Sequential message authentication code without random oracles

Bin Wang[a*] and Xiaojing Hong[b]

[a]No.196 West HuaYang Road, Information Engineering College of Yangzhou University, Yangzhou City, Jiangsu Province, 225127 P.R.China

[b]No.5 South YangZiJiang Road, Yangzhou City, Jiangsu Province, 225101 P.R. China

E-mail: jxbin76@yeah.net [a*]

Abstract: Katz et al. provided a generic transform to construct aggregate message authentication codes and imposed a lower bound on the length of one aggregate MAC tag. The lower bound shows that the required tag length is at least linear with the number of messages when fast verification such as constant or logarithmic computation overhead is required. Aggregate message authentication codes are useful in settings such as mobile ad-hoc networks where devices are resource-constrained and energy cost is at a premium. In this paper, we introduce the notion of sequential aggregate message authentication code (SAMAC). We present a security model for this notion under unforgeability against chosen message and verification query attack and construct an efficient SAMAC scheme by extending a number-theoretic MAC construction due to Dodis et al. We prove the security of our SAMAC scheme under the CDH assumption in the standard model. Our SAMAC scheme improves the lower bound with the help of the underlying algebraic structure. Performance analysis shows that our SAMAC scheme yields constant computation for the verifier as well as fixed length for one aggregate.

Keywords: Message authentication code; Sequential aggregate; CDH assumption; Chosen message and verification query attack

1. Introduction

Aggregate signature proposed by Boneh et al. [6] allows a collection of signatures from (possibly different) signers on (possibly different) messages to be aggregated into one short aggregate signature. The short aggregate signature can assure a verifier that all signers indeed signed for the collection of messages respectively. Aggregate signatures are very useful in

practical applications such as certificate chains, secure routing protocols [12] where compressing a collection of signatures will greatly improve application performance.

Sequential aggregate signature proposed in [15], a variant of aggregate signature, is distinct from aggregate signature in that aggregation operations are performed by each signer in turn and it is well suited for secure BGP protocol [17]. It is also shown in [4] how to construct proxy signature via sequential aggregate signature. Lu et al. presented a sequential aggregate signature scheme [14] based on Waters signature [19] secure in the standard model with the advantage that the costly computational overhead (e.g, pairing operations) of it is constant, while that of the aggregate signature scheme [6] is linear with the number of the messages.

Message authentication code, the private-key analog of digital signature, is one of the main primitives of interest in cryptography. Similarly, aggregate message authentication code is the private-key analog of aggregate signature. At present, there are relatively less research works on this topic [3,7] than those on aggregate signature. The formal study of it was not initiated until the work of Katz et al. [11]. It is pointed out in [11] that aggregate message authentication codes can also be used to improve efficiency of authenticated schemes that deal with aggregation of data in mobile ad-hoc networks [10,18], where devices are resource-constrained and the energy cost of communication is at a premium. They designed a security model that supports to aggregate MAC tags without any inherent order and constructed two secure aggregate message authentication code schemes. Their idea is to use XOR operations to compress MAC tags but the verification need re-compute MAC tags.

One drawback of the first aggregate message authentication code scheme (KL Scheme 1) in [11] is that a verifier has to re-compute l MAC operations to verify an aggregate MAC tag of fixed length on a set of l messages. Although the verifier (e.g., base station) may be more powerful than other nodes, it is still desirable to offload the processing time of the verifier as much as possible. To address this issue, they suggested a trade off between the length of one aggregate tag and the time required to verify the tag. In other words, one can verify a tag in constant time at the price of the tag of $O(l)$ length. The second aggregate message authentication code scheme (KL Scheme 2), which lies between the above two

extremes, achieves $O(\sqrt{l})$ verification time but the aggregate tag has length $O(\sqrt{l})$. Moreover, they also proved by their definition that the length of one aggregate MAC tag is at least $\Omega(l)$ when logarithmic $O(\log l)$ computation overhead for verification is required.

Traditionally, pseudorandom functions are used to construct message authentication codes. Pseudorandom function constructions can be relied upon symmetric-based primitives such as CBC MAC [2] or number-theoretic assumptions [16]. Although the former have the speed advantage, their security cannot be reduced to number-theoretic assumptions and they lack algebraic properties useful for some specific applications. On the other hand, most number-theoretic pseudorandom function constructions are comparably inefficient. It is an open question how to build an efficient MAC based on number-theoretic assumptions such as DDH assumption. To address this issue, Dodis et al. [8] presented a novel idea to design efficient probabilistic message authentication codes based on number-theoretic assumptions. Their idea is to remove the public verification functionality inherent in digital signatures to construct message authentication codes. In other words, only the secret key is used to verify validity of MAC tags.

We note that the generic construction of aggregation message authentication codes in [11] does not take potential algebraic structure into account and their result is obtained by assuming arbitrary order to aggregate MAC tags. Inspired by the ideas presented in [8], we aim to improve the lower bound [11] imposed on the length of one aggregate tag by constructing a sequential aggregation message authentication code (SAMAC) scheme based on number-theoretic assumptions. At first, it seems that we can simply use the technique in [8] to transform the sequential aggregate signature [14] secure in the standard model by removing the public verification functionality in order to yield a SAMAC scheme secure in the standard model. However, this method does not work because a signer in their scheme [14] need access the public keys of other users appearing in the aggregate-so-far in order to perform aggregate operation on it. The resulting scheme is not practical since the public key size required by their scheme [14] is quite large. For instance, the key of one user takes around 38KB to store if we assume a 160-bit collision resistant hash function. The authors [14] also suggested an open question to reduce the size of user keys for sequential aggregate signatures secure in the

standard model. On the other hand, it seems peculiar to keep the concept of public key in a symmetric primitive such as aggregate message authentication code.

We choose to extend the message authentication code scheme MAC_{BB} [8] which is based on Boneh-Boyen signature [5] to yield an efficient SAMAC scheme secure in the standard model. Performance analysis shows that our SAMAC scheme only requires a verifier to compute less than three exponentiation operations to verify validity of an aggregate tag, independent of l, the number of signers. In addition, the length of an aggregate tag consists of only three group elements.

The rest of this paper is organized as follows. At first, we describe the notations used in this paper and bilinear mappings in section 2. Syntax of SAMAC schemes is introduced in section 3. Katz et al. [11] considered security for aggregation message authentication codes under the notion of "unforgeability against chosen message attack(uf-cma)" [9]. It is reasonable in their setting because they mainly considered deterministic MACs constructed by primitives like block ciphers or hash functions. In this case, "unforgeability against chosen message (uf-cma) attack" is equivalent to "unforgeability against chosen message and verification query (uf-cmva) attack. For probabilistic MACs, some MAC constructions are only uf-cma secure, but not uf-cmva secure [1,13].

As our SAMAC scheme is probabilistic, we define security model for SAMAC schemes under the notion of "unforgeability against chosen message and verification query attack(uf-cmva)". In view of XOR operations used by the generic construction [11] make it difficult to take advantage of potential algebraic properties, we design a SAMAC scheme by extending the "algebraic message authentication code scheme" MAC_{BB} proposed in [8] and prove our SAMAC scheme to be secure under the CDH assumption in the standard model. Our scheme is proven to be selectively secure. That is, an adversary must commit the target forgery in advance. It is well known that a selectively secure cryptographic scheme implies a full secure one at the price of loss of a security degradation factor by guessing the forgery. Finally, we evaluate the performance of our SAMAC scheme to show that our scheme yields constant computation for a verifier as well as fixed length for an aggregate tag.

2. Preliminaries

2.1 Notation

We denote by λ a security parameter. If A is a randomized algorithm, then $y \leftarrow A(x_1, x_2, \cdots; r)$ means that A has input $x_1, x_2, \cdots$ and uniform random coins r, and the output of A is assigned to y. We use the notation $x \leftarrow_R S$ to mean "the element x is chosen with uniform probability from the set S".

2. 2 Bilinear pairing

Given a security parameter λ, an efficient algorithm $PG(1^\lambda)$ outputs (e, G, G_T, g, p), where G is a cyclic group of a prime order p generated by g. G_T is a cyclic group of the same order, and let $e: G \times G \to G_T$ be a efficiently computable bilinear function with the following properties:

1. Bilinear: $e(g^a, g^b) = e(g, g)^{ab}$, for all $a, b \in Z_p$.

2. Non-degenerate: $e(g, g) \neq 1_{G_T}$

3. Sequential aggregate message authentication codes

3.1 Syntax of sequential aggregate message authentication codes

A sequential aggregate message authentication code (SAMAC) scheme consists of the following algorithms:

(1) **Setup:** Given a security parameter λ, output a common parameter $cpar$.

(2) **KG:** Given the common parameter $cpar$, this key-generation algorithm generates a secret key k and its unique public reference id.

(3) **Mac:** Given a message m and a secret key k, this tagging algorithm outputs a tag by running $Mac((k, id), m) \to t$.

(4) **AMac:** Given a message m, a key-reference pair (k, id), an aggregate-so-far tag pt, a set of message-reference pairs $MI = \{(id_i, m_i)\}_{i=1}^{l}$, $l < n$. Let $I = \{id_i\}_{i=1}^{l}$ be the corresponding set of references derived from MI. n is a system parameter that serves as an upper bound on the number of signers. If $id \notin I$, this aggregate-tagging algorithm outputs

an aggregate tag: $AMac((k,id),MI,m,pt)\rightarrow t$. This description means that a signer is not allowed to sign twice in one aggregate tag.

(5) **Vrfy:** Given a set of message-reference pairs $MI=\{(id_i,m_i)\}_{i=1}^{l}$, a set of key-reference pairs: $K=\{(k_i,id_i)\}_{i=1}^{l}$, $l\le n$, and a tag t, this verification algorithm first verifies that no reference appears more than once, and outputs 1 to denote t is a valid tag on the messages in MI under the keys in K; otherwise it outputs 0. There is no inherent order enforced by this verification algorithm.

Correctness: For any message m, key-reference pair $(k,id)\leftarrow KG(1^{\lambda})$, an aggregate-so-far tag pt, and two sets: $MI=\{(id_i,m_i)\}_{i=1}^{l}$, $K=\{(k_i,id_i)\}_{i=1}^{l}$, $l<n$, if $Vrfy(MI,K,pt)\rightarrow 1$ and $AMac(k,MI,m,pt)\rightarrow t$, we require:

$$\Pr[Vrfy(MI\bigcup(m,id),K\bigcup(k,id),t)\rightarrow 1]=1$$

3.2 Security model for sequential aggregate message authentication codes

3.2.1 Unforgeability

Given a sequential aggregate message authentication code (SAMAC) scheme, consider the following game $\mathrm{Exp}_{\mathrm{SAMAC}}^{\text{uf-cmva}}(A,\lambda,n)$ between an adversary A and a game challenger S. Our description is obtained by extending the standard security notion of unforgeability under chosen message and verification query attack for MAC schemes.

Stage 1: The challenger S runs $KG(1^{\lambda})$ to obtain a challenge key-reference pair (k^*,id^*) and two empty sets KS,TS are also created, which are used to keep track of the registered keys as well as tag queries issued by the adversary respectively. Then A is provided with the reference id^*.

Stage 2: A can issue key queries, tag queries as well as verification queries which are handled as follows:

$Key(\cdot)$

$(k_i,id_i)\leftarrow KG(1^{\lambda})$;

$KS \leftarrow KS \bigcup \{id_i\}$; return k_i ;

$Tag(m, MI, pt)$

Let $MI = \{(id_i, m_i)\}_{i=1}^{l}$ and $I = \{id_i\}_{i=1}^{l}$ be derived from MI ;

If $l \geq n$, return $\perp$;

If $\exists id_i \in I$ but $id_i \notin KS$, return $\perp$;

If $id^* \notin I$ and each reference $id_i \in I$ appears only once,

$AMac((k^*, id^*), MI, m, pt) \rightarrow t$; $TS \leftarrow TS \bigcup \{m\}$;

return t ;

$V(MI, t)$

Let $MI = \{(id_i, m_i)\}_{i=1}^{l}$ and $I = \{id_i\}_{i=1}^{l}$ be derived from MI ;

If $l > n$, return $\perp$;

If $\exists id_i \in I$ but $id_i \notin KS$, return $\perp$;

Let $K = \{(k_i, id_i)\}_{i=1}^{l}$, where k_i is the secret key associated with $id_i \in I$;

If each reference $id_i \in I$ appears only once,

Run $Vrfy(MI, K, t) \rightarrow b$; return the bit b ;

Otherwise return $\perp$;

Finally, A outputs a forgery t^* on a set of message-reference pairs $MI^* = \{(id_i, m_i)\}_{i=1}^{l}$. Let $I^* = \{id_i\}_{i=1}^{l}$ be the set of references derived from MI^* . We require $id^* \in I^*, l \leq n$ and each reference in I^* appears only once. Let $K^* = \{(k_i, id_i)\}_{i=1}^{l}$, where k_i is the secret key associated with $id_i \in I^*$. A wins the game if the following hold:

(1) $Vrfy(MI^*, K^*, t^*) \to 1$;

(2) and the message m^* bound with id^* is not queried to the tag oracle by the adversary. That is, $m^* \notin TS$.

The advantage $\mathrm{Adv}^{\text{uf-cmva}}_{\text{SAMAC}}(A, \lambda, n)$ of A is defined to be the probability of A winning in this game. A SAMAC scheme is unforgeable if $\mathrm{Adv}^{\text{uf-cmva}}_{\text{SAMAC}}(A, \lambda, n)$ is negligible in λ for every PPT adversary A.

Remark: Selective security is defined by $\mathrm{Exp}^{\text{suf-cmva}}_{\text{SAMAC}}(A, \lambda, n)$, which is the same as $\mathrm{Exp}^{\text{uf-cmva}}_{\text{SAMAC}}(A, \lambda, n)$ except that the adversary must declare its target message m^* bound with id^* in advance.

4 Our SAMAC scheme

We adapt the message authentication code scheme $\mathrm{MAC}_{\mathrm{BB}}$ in [8] to yield a SAMAC scheme, which consists of the following algorithms:

1. Setup: Given the security parameter λ, $PG(1^\lambda)$ outputs (e, G, G_T, g, p). The common parameter is $cpar = (e, G, G_T, g, p)$. Messages to be signed are viewed as elements in Z_p.

2. KG: Picks a secret key $k = (x_1, x_2, y) \leftarrow_R Z_p^3$. The public reference for this key is a unique identifier id.

3. Mac: Given a secret key $k = (x_1, x_2, y)$ and a message m, a tag is generated as:

$$t = (t_1, t_2, t_3) = (U, g^{x_1 \cdot y} \cdot U^{x_1 \cdot m + x_2}, g^{x_1 \cdot m + x_2}),\ U \leftarrow_R G \setminus \{1_G\}$$

where 1_G is the identity over the group G. The seemingly redundant $g^{x_1 \cdot m + x_2}$ in this tag is introduced for fast aggregation.

4. AMac: Given a message m, a key-reference pair (k, id), an aggregate-so-far tag

pt, a set of message-reference pairs $MI = \{(id_i, m_i)\}_{i=1}^{l}$, $l < n$, an aggregate tag is generated as follows:

(1) Let $I = \{id_i\}_{i=1}^{l}$ be the corresponding set of references derived from MI. If $id \in I$, return $\perp$;

(2) Otherwise parse pt as (pt_1, pt_2, pt_3) and the secret key k as (x_1, x_2, y);

(3) If $pt_1 = 1_G$, return $\perp$;

(4) Compute $t_2 = pt_2 \cdot g^{x_1 \cdot y} \cdot (pt_1)^{x_1 \cdot m + x_2}, t_3 = pt_3 \cdot g^{x_1 \cdot m + x_2}$;

(5) Pick $\hat{u} \leftarrow_R Z_p$ and compute $t_1 = pt_1 \cdot g^{\hat{u}}, t_2 = t_2 \cdot (t_3)^{\hat{u}}$;

(6) Output $t = (t_1, t_2, t_3)$.

5. Vrfy: Given a set of message-reference pairs $MI = \{(id_i, m_i)\}_{i=1}^{l}$, a set of key-reference pairs: $K = \{(k_i, id_i)\}_{i=1}^{l}$, $l \le n$, and a tag t, the verification algorithm first verifies that each reference appears only once and proceeds as follows:

(1) Parse the tag t as (t_1, t_2, t_3), the secret key $k_i = (x_1^{(i)}, x_2^{(i)}, y^{(i)}), 1 \le i \le l$;

(2) Let $a_l = \sum_{i=1}^{l} x_1^{(i)} \cdot m_i + x_2^{(i)}, b_l = \sum_{i=1}^{l} x_1^{(i)} y^{(i)}$. Output 1 if

$$t_1 \neq 1_G \wedge t_2 = g^{b_l} \cdot (t_1)^{a_l} \wedge t_3 = g^{a_l} \quad (1)$$

We consider $pt = (1_G, 1_G, 1_G)$ as a valid tag under an empty set of signers. In this case, we take $a_0 = b_0 = 0$. In addition, it is not difficult to verify that the output of $AMac((k, id), \varnothing, m, pt = (1_G, 1_G, 1_G); \hat{u}) \to t$ over the randomness $\hat{u}$ is identically distributed to that of $Mac((k, id), m) \to t$.

Correctness: For any message m, key-reference pair $(k, id) \leftarrow KG(1^{\lambda})$, an aggregate-so-far tag pt, and two sets: $MI = \{(id_i, m_i)\}_{i=1}^{l}$, $K = \{(k_i, id_i)\}_{i=1}^{l}$, $1 \le l < n$, if $AVrfy(MI, K, pt) \to 1$, we proceed as follows:

(1) Parse the aggregate-so-far tag pt as (pt_1, pt_2, pt_3), the secret keys $k=(x_1, x_2, y)$, and $k_i = (x_1^{(i)}, x_2^{(i)}, y^{(i)}), 1 \le i \le l$;

(2) Let $a_l = \sum_{i=1}^{l} x_1^{(i)} \cdot m_i + x_2^{(i)}, b_l = \sum_{i=1}^{l} x_1^{(i)} y^{(i)}$. $AVrfy(MI, K, pt) \to 1$ means:

$(pt_1 = g^u) \neq 1_G \wedge pt_2 = g^{b_l} \cdot (pt_1)^{a_l} \wedge pt_3 = g^{a_l}$

Assume $\hat{u} \leftarrow_R Z_p$ be the randomness used by the algorithm **Amac**. The output $t = (t_1, t_2, t_3)$ of $AMac((k, id), MI, m, pt; \hat{u})$ is computed as follows:

At step (4) of **Amac**, $t_2 = pt_2 \cdot g^{x_1 \cdot y} \cdot (pt_1)^{x_1 \cdot m + x_2}, t_3 = pt_3 \cdot g^{x_1 \cdot m + x_2}$;

Plugging the expressions of pt_1, pt_2, pt_3 into t_2, t_3, we obtain:

$t_2 = (g^{b_l} \cdot (pt_1)^{a_l}) \cdot g^{x_1 \cdot y} \cdot (pt_1)^{x_1 \cdot m + x_2} = g^{b_l + x_1 \cdot y} \cdot (pt_1)^{a_l + x_1 \cdot m + x_2}$

$t_3 = g^{a_l} \cdot g^{x_1 \cdot m + x_2} = g^{a_l + x_1 \cdot m + x_2}$

At the re-randomization step (5) of **Amac**, $t_1 = pt_1 \cdot g^{\hat{u}} = g^{u + \hat{u}}, t_2 = t_2 \cdot (t_3)^{\hat{u}}$.

$t_2 = g^{b_l + x_1 \cdot y} \cdot (pt_1)^{a_l + x_1 \cdot m + x_2} \cdot (g^{a_l + x_1 \cdot m + x_2})^{\hat{u}}$

$= g^{b_l + x_1 \cdot y} \cdot (g^{u + \hat{u}})^{a_l + x_1 \cdot m + x_2} = g^{b_l + x_1 \cdot y} \cdot (t_1)^{a_l + x_1 \cdot m + x_2}$

The above result means $Vrfy(MI \bigcup (m, id), K \bigcup (k, id), t) \to 1$.

Although our scheme does not allow a signer to sign more than once in an aggregate tag, we can adopt the trick suggested in [section 3.2, 14] to handle this issue.

5 Security Analysis

Theorem 1: Assume there is an adversary A running in time t that can win the experiment $\mathrm{Exp}_{\mathrm{SAMAC}}^{\text{suf-cmva}}(A, \lambda, n)$ with probability ε. We can construct a simulator S that is able to solve the CDH problem over the group G equipped with bilinear mapping with probability $\varepsilon' \ge \varepsilon$, running in time $t' = t + O(q_k + q_t + q_V)$, where q_k, q_t, q_V are the number of key queries, tag queries and verification queries issued by A respectively.

Proof: The simulator S takes an instance $\{(g, g^x, g^y) : x, y \leftarrow_R Z_p\}$ of the CDH problem

over the group G equipped with bilinear mapping as input and simulates the environment of $\mathrm{Exp}_{\mathrm{SAMAC}}^{\text{suf-cmva}}(A,\lambda,n)$ for the adversary A as follows.

At first, S receives the target message m^* chosen by A. S picks $a \leftarrow_R Z_p$, the public reference id^* and implicitly defines the secret key $k^* = (x_1 \leftarrow x, x_2 \leftarrow x^*, y)$, where $x^* \leftarrow a - m^* \cdot x$. With this definition of the secret key k^*, a correct tag on a message m under k^* is of the form:

$$t = (g^u, g^{xy+u(a+x(m-m^*))}, g^{a+x(m-m^*)}) \tag{2}$$

The queries issued by A are handled as follows:

$Key(\cdot)$

Pick a secret key $k_i = (x_1^{(i)}, x_2^{(i)}, y^{(i)}) \leftarrow_R Z_p^{\,3}$. The reference for this key is a unique public identifier id.

$KS \leftarrow KS \bigcup \{id_i\}$; return k_i;

$Tag(m, MI, pt)$

If $m = m^*$, return $\perp$;

If MI is empty,

Pick $r \leftarrow Z_p$ and implicitly define $u = (-y)/(m - m^*) + r$.

As $u = 0$ implies $y = (m - m^*) \cdot r$, we can output the CDH solution $g^{xy} = (g^x)^{(m-m^*)\cdot r}$. In the following, assume $u \neq 0$. A correctly distributed tag is constructed as follows:

$t' = (g^u, g^{xy+u(a+x(m-m^*))}, g^{a+x(m-m^*)})$;

$g^u = (g^y)^{\frac{-1}{m-m^*}} \cdot g^r$, $g^{xy+u(a+x(m-m^*))} = (g^y)^{\frac{-a}{m-m^*}} \cdot g^{a\cdot r} \cdot (g^x)^{(m-m^*)\cdot r}$,

$g^{a+x(m-m^*)} = g^a \cdot (g^x)^{m-m^*}$;

$TS \leftarrow TS \bigcup \{m\}$;

return t';

Otherwise, let $MI = \{(id_i, m_i)\}_{i=1}^{l}$ and $I = \{id_i\}_{i=1}^{l}$ be derived from MI;

(1) If $l \geq n$, return $\perp$;

(2) If $\exists id_i \in I$ but $id_i \notin KS$, return $\perp$;

(3) If $id^* \notin I$ and each reference $id_i \in I$ appears only once, the simulator now constructs the rest of the required aggregate t by adding to t' the component on m_i computed by running $AMac((k_i, id_i), \cdot, m_i, \cdot)$ in turn, $1 \leq i \leq l$, where k_i is the secret key of id_i.

This can be accomplished because the simulator knows the registered secret keys. The output is identical to that of $AMac((k^*, id^*), MI, m, pt)$ on condition that the aggregate-so-far pt is correct. This is because the positions of signers appearing in an aggregate are interchangeable by the structure of the aggregate and our aggregation operation involves re-randomization such that the output is uniformly distributed over the correct aggregates. This is also the proof strategy adopted in [14].

(4) The simulator then proceeds as follows:

If $Vrfy(MI, K, pt) \rightarrow 1$, return t;

Otherwise,

If $pt_1 = 1_G$, return $\perp$;

If $pt_2 \neq g^{b_l} \cdot (pt_1)^{a_l}$, compute $t_2 = t_2 \cdot g^b, b \leftarrow_R Z_p^*$;

If $pt_3 \neq g^{a_l}$, compute $t_3 = t_3 \cdot g^c, c \leftarrow_R Z_p^*$

return t;

$V(MI, t)$

Let $MI = \{(id_i, m_i)\}_{i=1}^{l}$ and $I = \{id_i\}_{i=1}^{l}$ be derived from MI;

If $l > n$, return $\perp$;

If $\exists id_i \in I$ but $id_i \notin KS$, return $\perp$;

Let $K = \{(k_i, id_i)\}_{i=1}^{l}$, where k_i is the secret key associated with $id_i \in I$;

The case $id^* \notin I$ can be handled easily. Hence we only consider the case that $id^* \in I$. Assume that each reference $id_i \in I$ appears only once and without loss of generality $id^* = id_1$.

If $m_1 = m^*$, return $\perp$; otherwise proceed as follows:

(1) Parse the tag t as (t_1, t_2, t_3), $k_i = (x_1^{(i)}, x_2^{(i)}, y^{(i)}), 2 \le i \le l$;

(2) Let $a_{l-1} = \sum_{i=2}^{l} x_1^{(i)} \cdot m_i + x_2^{(i)}, b_{l-1} = \sum_{i=2}^{l} x_1^{(i)} y^{(i)}$;

(3) Compute $t_2^{/} = t_2 / (g^{b_l} \cdot (t_1)^{a_l}), t_3^{/} = t_3 / g^{a_l}$;

That is, we eliminate the parts involved with the registered secret keys to decide whether $(t_1, t_2^{/}, t_3^{/})$ is a correctly distributed tag on m_1 under id^*;

(4) Assume $(t_1, t_2^{/}, t_3^{/}) = (g^u, g^{xy+u(a+x(m_1-m^*))+z}, g^{a+x(m_1-m^*))+w})$;

(5) If $t_1 = 1_G$ or $t_3^{/} \neq g^a \cdot (g^x)^{m-m^*}$, outputs 0; otherwise assume $u = (-y)/(m_1 - m^*) + r$ for some unknown $r \in Z_p$;

(6) Plugging the expression of u into $(t_1, t_2^{/})$, we have

$t_1 = (g^y)^{\frac{-1}{m_1 - m^*}} \cdot g^r$;

$t_2^{/} = g^{xy+u(a+x(m_1-m^*))+z} = g^{x \cdot r \cdot (m_1 - m^*)} \cdot g^{a \cdot r} \cdot (g^y)^{\frac{-a}{m_1 - m^*}} \cdot g^z$;

Compute $t_1^{(m_1 - m^*)} / g^y = g^{r(m_1 - m^*)}$, $t_2^{/} / (t_1)^a = g^{x \cdot r \cdot (m_1 - m^*)} \cdot g^z$.

(7) If $e(g, t_2^{/} / (t_1)^a) = e(g^x, t_1^{(m_1 - m^*)} / g^y)$, the simulator outputs 1 since this implies $z = 0$; otherwise it outputs 0.

Finally, A outputs a forgery t^* on a set of message-reference pairs

$MI^* = \{(id_i, m_i)\}_{i=1}^{l}$. Let $I^* = \{id_i\}_{i=1}^{l}$ be the set derived from MI^* . We require $id^* \in I^*, l \le n$ and each reference appears only once. Assume without loss of generality $id^* = id_1$ and $m_1 = m^*$ since the positions of signers are interchangeable by the structure of an aggregate.

Let $K^* = \{(k_i, id_i)\}_{i=1}^{l}$, where k_i is the secret key associated with $id_i \in I^*$.

(1) Parse $t^* = (t_1^*, t_2^*, t_3^*), k_i = (x_1^{(i)}, x_2^{(i)}, y^{(i)}), 2 \le i \le l$;

(2) If $m^* \notin TS$, proceed as follows:

Let $a_{l-1} = \sum_{i=2}^{l} x_1^{(i)} \cdot m_i + x_2^{(i)}, b_{l-1} = \sum_{i=2}^{l} x_1^{(i)} y^{(i)}$;

The adversary A wins if $Vrfy(MI^*, K^*, t^*) \to 1$, which means:

$t_2^* = g^{xy} \cdot g^{b_{l-1}} \cdot (t_1^*)^{a_{l-1}} \cdot (t_1^*)^{x \cdot m^* + x^*} = g^{xy} \cdot g^{b_{l-1}} \cdot (t_1^*)^{a_{l-1}} \cdot (t_1^*)^{a}$.

(3) The simulator outputs the CDH solution as:

$$g^{xy} = t_2^* / (g^{b_{l-1}} \cdot (t_1^*)^{a + a_{l-1}})$$

The correctness of the solution can be checked as follows:

$$e(g^x, g^y) \stackrel{?}{=} e(g, t_2^* / (g^{b_{l-1}} \cdot (t_1^*)^{a + a_{l-1}}))$$

Let E be the event "the implicitly defined randomness $u = 0$ when answering one tag query" and F be the event "the simulator S solves the CDH problem". As the simulator S provides a perfect simulation for the adversary A on condition that $\overline{E}$ occurs, the probability ε' that S solves the CDH problem can be estimated as follows:

$$\varepsilon' = \Pr[F \mid E]\Pr[E] + \Pr[F \mid \overline{E}]\Pr[\overline{E}]$$

$$= 1 \cdot \Pr[E] + \varepsilon \Pr[\overline{E}] = \varepsilon + \Pr[E] \cdot (1 - \varepsilon) \ge \varepsilon$$

Although our scheme is selectively secure, it is well known that a selectively secure cryptographic scheme implies a full secure one at the price of loss of a security degradation

factor by guessing the forgery. In addition, the typical message transmitted in the mobile ad-hoc networks may be very short (e.g. 16 bit status information [11]).

6. Performance Analysis

In this section, we evaluate the performance of our sequential aggregation message authentication code scheme and the two aggregate message authentication code schemes [11] in terms of the computational cost to verify an aggregate and the length of an aggregate. The result is stated in Table 1, where l is the number of signers. Exp, MAC denote one exponentiation operation and one underlying MAC operation respectively. $|T|$ represents the length of a MAC tag T produced by the underlying MAC scheme. $|G|$ represents the length of one element of the group G. The optimized processing time of one multi-exponentiation $g^a h^b$ is less than that of 1.5 Exp.

The KL scheme 1 in [11] requires a verifier to re-compute l MAC operations when the length of an aggregate is fixed. On the other hand, they show constant verification time can be achieved at the price of a tag of length $l|T|$. They also demonstrated that the length of one aggregated tag is at least $\Omega(l)$ when logarithmic computation overhead is required for verification. To address this issue, they presented an scheme, which lies between the above two extremes, to obtain $O(\sqrt{l})$ verification time and the aggregate tag has length $|T|O(\sqrt{l})$.

As their result is obtained by assuming to aggregate MAC tags in any order and XOR operations used by their generic construction make it difficult to take advantage of potential algebraic properties, we present a sequential aggregated MAC scheme based on the number-theoretic CDH assumption to improve the lower bound in [11]. As a result, we obtain constant verification time as well as fixed length for one aggregate as shown in the table.

7. Conclusion

The formal study of aggregate message authentication code was not initiated until the work of Katz et al. [11]. Aggregate message authentication codes are useful for improving the efficiency of authenticated schemes that deal with aggregation of data in mobile ad-hoc networks. They also demonstrated that the length of one aggregate MAC tag is at least linear when logarithmic verification time is required. In this paper, we introduce the notion of sequential aggregate message authentication code and provide an efficient SAMAC construction by extending the number-theoretic message authentication code scheme MAC_{BB} [8]. The selective security of our SAMAC scheme is proved under the CDH assumption without random oracles. The algebraic structure underlying our construction helps to yield constant verification time and fixed length for one aggregate.

Acknowledgement

This work is supported by Natural Science Foundation of Higher Education Institutions, in Jiangsu Province office of education, P.R. China (Grant No. 10KJD520005).

References

[1] M. Bellare, O. Goldreich, and A. Mityagin. The power of verification queries in message authentication and authenticated encryption. Cryptology ePrint Archive, Report 2004/309, 2004. http://eprint.iacr.org/.

[2] M. Bellare, K. Pietrzak, P.Rogaway. Improved security analyses for CBC MACs. CRYPTO 2005, LNCS 3621 (2005), 527 – 545.

[3] R. Bhaskar, J. Herranz, and F. Laguillaumie. Aggregate designated verifier signatures and application to secure routing. Intl. J. Security and Networks 2(3/4) (2007) 192–201.

[4] A. Boldyreva, A. Palacio, B. Warinschi, Secure proxy signature schemes for delegation of signing rights, Journal of Cryptology, 25(1), (2012) 57-115

[5] Boneh, D., Boyen, X., Short signatures without random oracles. EuroCrytpt 2004, LNCS, 3027(2004), 56–73.

[6] D. Boneh, C.Gentry, B. Lynn, H. Shacham, Aggregate and verifiably encrypted signatures from bilinear maps, EuroCrytpt 2003, LNCS 2656 (2003) 416-432.

[7] H. Chan, A. Perrig, and D. Song. Secure hierarchical in-network aggregation in sensor networks. In ACM CCS ’06: 13th ACM Conf. on Computer and Communications Security, pages 278–287. ACM Press, 2006.

[8] Y. Dodis, E. Kiltz, K. Pietrzak, D.Wichs, Message authentication, revisited , EuroCrytpt 2012, LNCS 7237 (2012) 355-374.

[9] S. Goldwasser, S.Micali and R. Rivest, A digital signature scheme secure against adaptive chosen-message attacks, SIAM Journal of Computing, 17(2) (1988) 281-308.

[10] L. Hu and D. Evans. Secure aggregation for wireless networks. In Workshop on Security and Assurance in Ad-Hoc Networks (2003), 384–394.

[11] J. Katz, Y. Lindell, "Aggregate message authentication codes", IET Information security, to be published

[12] S.Kent, C.Lynn, K.Seo, Secure border gateway protocol (secure-BGP), IEEE J.Sel.Areas Commun., 18(4) (2000), 582-592.

[13] E. Kiltz, K. Pietrzak, D.Cash, A. Jain, and D. Venturi, Efficient authentication from hard learning problems. EUROCRYPT 2011, LNCS 6632 (2011), 7−26.

[14] S. Lu, R. Ostrovsky, A. Sahai, H. Shacham, B.Waters, Sequential aggregate signatures, multisignatures and verifiably encrypted signatures without random oracles, Journal of Cryptology, 26(2), (2013) 340-373

[15] A. Lysyanskaya, S. Micali, L. Reyzin, H. Shacham, Sequential aggregate signatures from trapdoor permutation, EuroCrytpt 2004, LNCS 3027 (2004) 74-90.

[16] M. Naor, O. Reingold. Number-theoretic constructions of efficient pseudo-random functions. In: 38th Annual Symposium on Foundations of Computer Science, IEEE Computer Society Press (1997) 458–467.

[17] D. Nicol, S. Smith, M. Zhao, Evaluation of efficient security for BGP route announcements using parallel simulation, Simul.Model.Pract.Theory, 12(2004) 187–216

[18] B. Przydatek, D. Song, and A. Perrig, SIA: Secure information aggregation in sensor networks, 1st ACM Conference on Embedded Networked Sensor Systems (SenSys) 2003, 255–265.

[19] B. Waters, Efficient identity-based encryption without random oracles, EuroCrypt 2005, LNCS 3494(2005) 114–127.

Table 1. Performance comparison

Scheme	Computation cost to verify an aggregate tag	Length of an aggregate tag
KL scheme 1 [11]	l MAC	$\|T\|$
KL scheme 2 [11]	$O(\sqrt{l})$ MAC	$O(\|T\|\sqrt{l})$
The proposed SAMAC scheme	<2.5Exp	$3\|G\|$