Information-Theoretically Secure Aggregate Authentication Code: Model, Bounds, and Constructions

Asato Kubai, Junji Shikata, Yohei Watanabe

Information-Theoretically Secure Aggregate Authentication Code: Model, Bounds, and Constructions

Asato Kubai, Junji Shikata, and Yohei Watanabe

Graduate School of Environment and Information Sciences,
Yokohama National University, Japan
{shikata, watanabe-yohei-xs}@ynu.ac.jp

Abstract. In authentication schemes where many users send authenticated messages to a receiver, it is desirable to aggregate them into a single short authenticated message in order to reduce communication complexity. In this paper, in order to realize such a mechanism in information-theoretic security setting, we first propose aggregate authentication codes. Specifically, we newly propose a model and a security definition for aggregate authentication codes. We also show tight lower bounds on sizes of entities' secret-keys and (aggregated) tags. Furthermore, we present optimal (i.e., most efficient) constructions for aggregate authentication codes.

1 Introduction

1.1 Background

The security of most of present cryptographic systems is based on the assumption of difficulty of computationally hard problems such as the integer factoring problem or the discrete logarithm problem in finite fields or elliptic curves. However, taking into account recent rapid development of algorithms and computer technologies, such a system based on the assumption of difficulty of computationally hard problems might not maintain sufficient long-term security. In fact, it is known that quantum computers can easily solve the factoring and discrete logarithm problems. From these aspects, it is necessary and interesting to consider cryptographic techniques whose security does not depend on any computationally hard problems, especially for the long-term security.

Authentication is one of the fundamental and important functionalities in cryptography. Many papers in modern cryptography focus on constructing secure authentication schemes so that they are as efficient as possible, especially, in terms of communication complexity (e.g., size of authentication data including a MAC (message authentication code) or a digital signature sent via a public channel) and storage space (i.e., memory-size of users to keep secret-key data) in addition to time complexity (i.e., running time required for executing algorithms in the schemes). In authentication schemes where many users send authenticated

messages to a receiver (e.g., see the *multisender authentication code* [4],[9]), it is desirable to aggregate them into a single short authenticated message, since communication complexity required can be reduced. In particular, this mechanism is useful in the applications in which data-size per transmission in a channel is restricted (e.g., wireless communication). To solve this problem, Boneh et al. [2] proposed the first *aggregate signature* scheme. Unlike *multi-signatures* (e.g., [10]) in which a set of users all sign the same message and the result is a single signature, this is a scheme for combining various signatures from different signers on different messages into a single short signature. Since Boneh et al. gave a formal definition of aggregate signatures in [2], various research on aggregate signatures has been done based on computational security: for instance, sequential aggregate signatures (e.g., [8],[7]) for certificate chains and certificateless aggregate signatures (e.g., [3]). We note that the first aggregate signature scheme [2] is restricted in the sense that only aggregation of distinct messages is allowed. For lifting the restriction, Bellare et al. [1] proposed *unrestricted aggregate signatures*. On the other hand, as these protocols mentioned above are specific to the public-key setting, Katz et al. [6] proposed the *aggregate message authentication code* (aggregate MAC for short) which is specific to the shared-key (secret-key) setting. The aggregate MAC is a useful tool for the problem of authenticated communication in a mobile ad-hoc network where communication is considered as a highly expensive resource.

To the best of our knowledge, there is no paper which reports on the study of information-theoretically secure aggregate authentication schemes. Therefore, in this paper we newly introduce and realize *aggregate authentication codes* (aggregate A-codes for short) with information-theoretic security.

1.2 Our Contribution

The authentication code (A-code for short) (e.g., see [12]) is one of the fundamental cryptographic primitives with information-theoretic security. In the model of the traditional A-code, a single sender transmits an authenticated message to a single receiver. In the scenario where there are many entities and they communicate each other, it is not practical to use the A-code for every possible pair of entities. In particular, in authentication schemes where many users send authenticated messages to a receiver, it is desirable to aggregate them into a single short authenticated message in order to reduce communication complexity. Therefore, we study information-theoretically secure aggregate A-codes. Specifically, our contribution is as follows.

- We propose a model and formalization of security for aggregate A-codes in information-theoretic security setting. In our model, aggregation of not only distinct messages but also same messages is possible, and in this sense our scheme is unrestricted;
- We also derive tight lower bounds on entities' memory-sizes and (aggregated) tags required for aggregate A-codes; and

– We present two kinds of constructions, generic and direct ones. Our generic construction of aggregate A-codes is very simple: aggregate A-codes can be constructed from only traditional A-codes. Since the generic construction does not lead to an optimal construction of aggregate A-codes, we also propose a direct construction which is optimal (i.e., most efficient).

The rest of our paper is organized as follows. In Section 2, we propose a formal model and formalization of security for aggregate A-codes with information-theoretic security. In Section 3, we derive tight lower bounds on entities' memory-sizes and (aggregated) tags required for aggregate A-codes. Section 4 is devoted to present generic and direct constructions. Finally, in Section 5 we give concluding remarks of the paper.

Throughout this paper, we use the following notation. For any finite set $\mathcal{Z}$, let $\mathcal{P}(\mathcal{Z}) := \{Z \subset \mathcal{Z}\}$ be the family of all subsets of $\mathcal{Z}$. Also, for any finite set $\mathcal{Z}$ and any non-negative integer z, let $\mathcal{P}(\mathcal{Z}, z) := \{Z \subset \mathcal{Z} \mid |Z| \leq z\}$ be the family of all subsets of $\mathcal{Z}$ whose cardinality is less than or equal to z.

2 The Model and Security Definition

In this section, we introduce a model and a security definition of aggregate A-codes, based on those of aggregate MACs with computational security and those of traditional A-codes with information-theoretic security.

2.1 The Model

We show a model of aggregate A-codes. For simplicity, we assume that there is a trusted authority whose role is to generate and to distribute secret-keys of entities. We call this model the *trusted initializer model* as in [11]. In aggregate A-codes, there are $n + 2$ entities, n senders $T_1, T_2, \ldots, T_n$, a receiver R and a trusted initializer TI, where n is a positive integer. In this paper, we assume that the identity of each sender T_i is also denoted by T_i, and the receiver is honest in the model. Our model of aggregate A-codes is almost the same as that of aggregate MACs [6] except for considering the trusted initializer in our model. For simplicity, we consider a *one-time model* of aggregate A-codes, in which each sender is allowed to generate an authenticated message and aggregation is allowed to be executed at most only once.

Informally, an aggregate A-code is executed as follows. In the initial phase, TI generates secret-keys on behalf of T_i $(1 \leq i \leq n)$ and the receiver R. After distributing these keys via secure channels, TI deletes them in his memory. Any set of senders participate in the protocol, and in this paper we call them *active senders* for convenience. Each active sender generates a tag (or an authenticator) by using his secret-key. These tags can be aggregated into a single short tag, which we call an *aggregated tag*, without any secret-key. After the aggregated tag is transmitted, the receiver can check the validity of the aggregated tag by using his verification-key.

Formally, we give a definition of aggregate A-codes as follows.

Definition 1 (Aggregate A-code). An *aggregate authentication code* (*aggregate A-code* for short) Π involves $n+2$ entities, TI, $T_1, T_2, \ldots, T_n$ and R, and consists of a four-tuple of algorithms (*KGen*, $Auth_i$, *Agg*, *Vrfy*) with five spaces, $\mathcal{M}, \mathcal{A}, \mathcal{A}^*, \mathcal{E}_T$, and $\mathcal{E}_R$, where all of the above algorithms except *KGen* are deterministic and all of the above spaces are finite. In addition, Π is executed with four phases as follows.

0. **Notation.**
- *Entities*: TI is a trusted initializer, T_i $(1 \leq i \leq n)$ is a sender and R is a receiver. Let $\mathcal{T} := \{T_1, T_2, \ldots, T_n\}$ be the set of all senders, and let $\mathcal{S} := \{T_{i_1}, \ldots, T_{i_j}\} \in \mathcal{P}(\mathcal{T})$ be a set of active senders with $|\mathcal{S}| \geq 1$.
- Spaces: $\mathcal{M}$ is a set of possible messages, $\mathcal{A}$ is a set of possible tags (or authenticators) generated by each $T_i \in \mathcal{S}$. For any $\mathcal{S} \in \mathcal{P}(\mathcal{T})$ with $l = |\mathcal{S}|$, let $\mathcal{M}^{(l)} := \bigcup_{j=1}^{l} (\mathcal{M} \times \mathcal{S})^j$ and $\mathcal{A}^{(l)} := \bigcup_{j=1}^{l} \mathcal{A}^j$. $\mathcal{A}^*$ is a set of possible aggregated tags. Also, $\mathcal{E}_i$ is a set of possible T_i's secret-keys and $\mathcal{E}_R$ is a set of possible verification-keys. For simplicity, we assume $\mathcal{E}_1 = \mathcal{E}_2 = \cdots = \mathcal{E}_n$.
- Algorithms: *KGen* is a key generation algorithm which on input a security parameter 1^k, outputs each sender's secret-key and a receiver's verification-key. $Auth_i$: $\mathcal{M} \times \mathcal{E}_i \to \mathcal{A}$ is T_i's authentication algorithm. For every $1 \leq l \leq n$, Agg_l: $\mathcal{A}^{(l)} \to \mathcal{A}^*$ is an aggregation algorithm which compresses l tags into a single tag, and $Vrfy_l$: $\mathcal{M}^{(l)} \times \mathcal{A}^* \times \mathcal{E}_R \to \{true, false\}$ is a verification algorithm for l messages. In the following, we will briefly write *Agg* and *Vrfy* for Agg_l and $Vrfy_l$, respectively, if l is clear from the context.

1. **Key Generation and Distribution.** In the initial phase, by using *KGen* TI generates a secret-key $e_i \in \mathcal{E}_i$ for T_i $(i = 1, 2, \ldots, n)$ and a verification-key $e_v \in \mathcal{E}_R$ for R. These keys are distributed to corresponding entities via secure channels. After distributing these keys, TI deletes them from his memory. And, T_i and R keep their keys secret, respectively.
2. **Authentication.** For a message $m_i \in \mathcal{M}$, each $T_i \in \mathcal{S}$ can compute a tag $tag_i = Auth_i(m_i, e_i) \in \mathcal{A}$ by using his secret-key e_i.
3. **Aggregation.** Let $M := ((m_{i_1}, T_{i_1}), \ldots, (m_{i_j}, T_{i_j}))$. Any user can compute an aggregated tag $tag = Agg(tag_{i_1}, \ldots, tag_{i_j})$ by using only tags. [1] Then, the user transmits (M, tag) to R via an insecure channel.
4. **Verification.** Suppose that R has received (M, tag) via an insecure channel. R checks the validity of tag by a verification-key e_v: if $Vrfy(M, tag, e_v) = true$, then R accepts (M, tag) as valid, and rejects it otherwise.

In the model of aggregate A-codes, the following correctness condition is required to hold: for all possible $m_i \in \mathcal{M}$, $e_i \in \mathcal{E}_i$ $(1 \leq i \leq n)$, and $e_v \in \mathcal{E}_R$, if $tag_i = Auth_i(m_i, e_i)$ for each $T_i \in \mathcal{S}$ and $tag = Agg(tag_{i_1}, \ldots, tag_{i_j})$, it holds that

$$Vrfy(M, tag, e_v) = true.$$

[1] Not only any sender, but also anyone who does not have a secret-key can compute an aggregated tag, since this algorithm is executed without any secret-key. And also, in this model we represent multiple messages and tags as sequences for convenience, however, unlike [7], [8], our scheme is not sequential one (i.e., the order of messages and tags is not important).

The above requirement implies that any legal aggregated tag can be accepted without any error if entities correctly follow the specification of aggregate A-codes.

In addition, we formally define an *aggregation rate* which measures efficiency of compression for aggregated tags.

Definition 2 (Aggregation rate). Let Π be an aggregate A-code. An aggregation rate in Π is defined by

$$\gamma := \frac{\log |\mathcal{A}^*|}{\log |\mathcal{A}|}.$$

Note that it is natural to assume $|\mathcal{A}^*| \geq |\mathcal{A}|$, which implies $\gamma \geq 1$. On the other hand, considering the trivial aggregate A-code where the algorithm Agg_l is the identity mapping (i.e., an aggregated tag consists of concatenation of l multiple-tags) for any $1 \leq l \leq n$, we have $\gamma \leq l (\leq n)$. Therefore, for any $\mathcal{S} \in \mathcal{P}(\mathcal{T})$, it holds that

$$1 \leq \gamma \leq |\mathcal{S}| \ (\leq n).$$

An interesting case is $\gamma \ll |\mathcal{S}|$ even for large $\mathcal{S}$, and it is ideal when $\gamma \approx 1$ not depending on the size $|\mathcal{S}|$. In this paper, we will actually propose construction of aggregate A-codes which satisfies $\gamma = 1$ with having enough security (a formal security definition is given in the next subsection).

2.2 Security Definition

We formalize a security definition for aggregate A-codes. Let ω $(< n)$ be the maximum number of possible corrupted senders. For a set of corrupted senders (i.e., a colluding group) $W = \{T_{l_1}, T_{l_2}, \ldots, T_{l_j}\} \in \mathcal{P}(\mathcal{T}, \omega)$, $\mathcal{E}_W := \mathcal{E}_{l_1} \times \mathcal{E}_{l_2} \times \cdots \times \mathcal{E}_{l_j}$ denotes the set of possible secret-keys held by W.

In aggregate A-codes, we consider *impersonation attacks* and *substitution attacks*. The formalization of security notions for the above two kinds of attacks is given as follows.

Definition 3 (Security). Let Π be an aggregate A-code with an aggregation rate γ. For any set of active senders $\mathcal{S} \in \mathcal{P}(\mathcal{T})$ and any set of colluding groups $W \in \mathcal{P}(\mathcal{T}, \omega)$ such that $\mathcal{S} - W \neq \emptyset$, Π is said to be $(n, \omega, \epsilon, \gamma)$-*one-time secure*, if $\max(P_I, P_S) \leq \epsilon$, where P_I and P_S are defined as follows.

a) *Impersonation attacks.* The adversary who corrupts at most ω senders tries to generate a fraudulent pair of messages and aggregated tags (M, tag) such that (M, tag) is accepted by the receiver R. The success probability of this attack denoted by P_I is defined as follows: We define $P_I(\mathcal{S}, W)$ by

$$P_I(\mathcal{S}, W) = \max_{e_W \in \mathcal{E}_W} \max_{(M,tag)} \Pr(\mathit{Vrfy}(M, tag, e_v) = \mathit{true} \mid e_W).$$

The probability P_I is defined as $P_I := \max_{\mathcal{S},W} P_I(\mathcal{S}, W)$.

b) *Substitution attacks.* Let $\mathcal{S} = \{T_{i_1}, \ldots, T_{i_j}\}$. The adversary corrupts at most ω senders, and after observing valid pairs of messages and tags generated by $\mathcal{S}$, $((m_{i_1}, tag_{i_1}), \ldots, (m_{i_j}, tag_{i_j}))$, the adversary tries to generate a fraudulent pair of messages and aggregated tags, (M', tag'), that has not been legally generated by $\mathcal{S}$ but will be accepted by the receiver R such that $(M, tag) \neq (M', tag')$, where $M = ((m_{i_1}, T_{i_1}), \ldots, (m_{i_j}, T_{i_j}))$ and tag is an aggregated tag of M. The success probability of this attack denoted by P_S is defined as follows: We define $P_S(\mathcal{S}, W)$ by

$$P_S(\mathcal{S}, W) = \max_{e_W \in \mathcal{E}_W} \max_{((m_{i_1}, T_{i_1}), \ldots, (m_{i_j}, T_{i_j}))} \max_{(M', tag') \neq (M, tag)} \Pr(\mathit{Vrfy}(M', tag', e_v) = \mathit{true} \mid e_W, ((m_{i_1}, T_{i_1}, tag_{i_1}), \ldots, (m_{i_j}, T_{i_j}, tag_{i_j}))).$$

The probability P_S is defined as $P_S := \max_{\mathcal{S}, W} P_S(\mathcal{S}, W)$.

3 Lower Bounds

In this section, we derive lower bounds on success probabilities of attacks and memory-sizes required for $(n, \omega, \epsilon, \gamma)$-one-time secure aggregate A-codes. Let $\mathcal{MA}_i := \{(m_i, tag_i) \in \mathcal{M} \times \mathcal{A} \mid \mathit{Auth}_i(m_i, e_i) = tag_i \text{ for some } e_i \in \mathcal{E}_i\}$ be a set of possible pairs of messages and tags such that each element of the set can be generated by the sender T_i. And let $\mathcal{MA}^* := \{(m_{i_1}, \ldots, m_{i_j}, tag) \in \mathcal{M}^j \times \mathcal{A}^* \mid \mathit{Agg}(tag_{i_1}, \ldots, tag_{i_j}) = tag \wedge \mathit{Auth}_i(m_i, e_i) = tag_i \text{ for some } e_i \in \mathcal{E}_i (1 \leq i \leq j)\}$ be a set of possible pairs of messages and aggregated tags such that each element of the set can be generated by the senders $\mathcal{S} = \{T_{i_1}, T_{i_2}, \ldots, T_{i_j}\}$. Furthermore, let MA_i, MA^*, E_i, E_v, and E_W be random variables which take values in $\mathcal{MA}_i$, $\mathcal{MA}^*$, $\mathcal{E}_i$, $\mathcal{E}_R$, and $\mathcal{E}_W$, respectively. And also, let $(MA^*, \tilde{MA}^*)$ be a joint random variable which takes values in the set $\mathcal{MA}^* \times \mathcal{MA}^*$ such that $MA^* \neq \tilde{MA}^*$.

We assume that there exists the following mapping in the model of aggregate A-codes:

$$\pi : \mathcal{E}_R \to \mathcal{E}_1 \times \cdots \times \mathcal{E}_n.$$

Note that this assumption is not so strong, since we will actually see this mapping in our simple construction in Section 4.2. Then, we can derive lower bounds on success probabilities of attacks as follows.

Theorem 1. For any $i \in \{1, 2, \ldots, n\}$, any set of active senders $\mathcal{S} = \{T_{i_1}, \ldots, T_{i_j}\} \in \mathcal{P}(\mathcal{T})$ and any set of colluding groups $W \in \mathcal{P}(\mathcal{T}, \omega)$ such that $\mathcal{S} - W \neq \emptyset$, it holds that

1. $\log P_I(\mathcal{S}, W) \geq -I(MA^*; E_v \mid E_W)$,
2. $\log P_S(\mathcal{S}, W) \geq -I(\tilde{MA}^*; E_v \mid E_W, MA_{i_1}, \ldots, MA_{i_j})$.

Proof Sketch. The proof can be shown in a way similar to that of Theorem 1 in [5]. Here, we show an outline of a proof of the first inequality.

We define a characteristic function $\mathcal{X}_I$ as follows.

$$\mathcal{X}_I((M,tag),e_v,e_W) = \begin{cases} 1 \text{ if } \mathit{Vrfy}(M,tag,e_v) = \mathit{true} \\ \quad \wedge \Pr((M,tag),e_v,e_W) \neq 0, \\ 0 \ \mathit{otherwise}. \end{cases}$$

Then, from Definition 3, we can express $P_I(\mathcal{S},W)$ as

$$P_I(\mathcal{S},W) = \max_{(M,tag)} \max_{e_W} \sum_{e_v} \mathcal{X}_I((M,tag),e_v,e_W) \Pr(e_v \mid e_W).$$

By a way similar to the proof of Theorem 1 in [5], we have $P_I(\mathcal{S},W) \geq 2^{-I(MA^*;E_v|E_W)}$. Similarly, the second inequality can also be proved. □

We next show lower bounds on memory-sizes of entities in aggregate A-codes.

Theorem 2. Let Π be an $(n,\omega,\epsilon,\gamma)$-one-time secure aggregate A-code. Let $q := \epsilon^{-1}$. Then, for any $i \in \{1,2,\ldots,n\}$, we have

$$(i)\ |\mathcal{E}_i| \geq q^2, \quad (ii)\ |\mathcal{E}_R| \geq q^{2(\omega+1)}, \ (iii)\ |\mathcal{A}| \geq q, \quad (iv)\ |\mathcal{A}^*| \geq q.$$

Proof. In order to complete the proof of Theorem 2, we show the following lemmas.

Lemma 1. For arbitrary $i \in \{1,2,\ldots,n\}$, let $\mathcal{S}_i = \{T_{i_1},\ldots,T_{i_j}\} \in \mathcal{P}(\mathcal{T})$ and $W \in \mathcal{P}(\mathcal{T},\omega)$ such that $\mathcal{S}_i - W = \{T_i\}$. Then, we have

$$\log P_I(\mathcal{S}_i,W) \geq -I(MA^*;E_i \mid E_W),$$
$$\log P_S(\mathcal{S}_i,W) \geq -H(E_i \mid E_W, MA_{i_1},\ldots,MA_{i_j}).$$

Proof. For the first inequality, we get

$$\begin{aligned} I(MA^*;E_v \mid E_W) &= H(MA^* \mid E_W) - H(MA^* \mid E_v,E_W) \\ &= H(MA^* \mid E_W) - H(MA^* \mid E_i,E_W) \qquad (1) \\ &= I(MA^*;E_i \mid E_W), \end{aligned}$$

where (1) follows from the following equality: from the mapping π,

$$H(A^*|E_v,E_W,M) = H(A^*|E_i,E_W,M) = 0.$$

Hence, by Theorem 1, we have $\log P_I(\mathcal{S}_i,W) \geq -I(MA^*;E_i \mid E_W)$.

For the second inequality, we derive

$$\begin{aligned} & I(\tilde{M}\tilde{A}^*;E_v \mid E_W, MA_{i_1},\ldots,MA_{i_j}) \\ &= H(\tilde{M}\tilde{A}^* \mid E_W, MA_{i_1},\ldots,MA_{i_j}) \\ &\qquad -H(\tilde{M}\tilde{A}^* \mid E_v,E_W, MA_{i_1},\ldots,MA_{i_j}) \\ &= H(\tilde{M}\tilde{A}^* \mid E_W, MA_{i_1},\ldots,MA_{i_j}) \\ &\qquad -H(\tilde{M}\tilde{A}^* \mid E_i,E_W, MA_{i_1},\ldots,MA_{i_j}) \qquad (2) \\ &= I(\tilde{M}\tilde{A}^*;E_i \mid E_W, MA_{i_1},\ldots,MA_{i_j}) \\ &\leq H(E_i \mid E_W, MA_{i_1},\ldots,MA_{i_j}). \end{aligned}$$

where (2) follows from the following equality: from the mapping π,

$$\begin{aligned}&H(\tilde{A^*}|E_v, E_W, MA_{i_1}, \ldots, MA_{i_j}, \tilde{M})\\ &= H(\tilde{A^*}|E_i, E_W, MA_{i_1}, \ldots, MA_{i_j}, \tilde{M}) = 0.\end{aligned}$$

Hence, by Theorem 1, we have $\log P_S(\mathcal{S}_i, W) \geq -H(E_i \mid E_W, MA_{i_1}, \ldots, MA_{i_j})$. □

Lemma 2. $|\mathcal{E}_i| \geq q^2$ for any $i \in \{1, 2, \ldots, n\}$.

Proof. For arbitrary $i \in \{1, 2, \ldots, n\}$, let $W \in \mathcal{P}(\mathcal{T}, \omega)$ and $\mathcal{S}_i = \{T_{i_1}, \ldots, T_{i_j}\} \in \mathcal{P}(\mathcal{T})$ such that $\mathcal{S}_i - W = \{T_i\}$. Then, we have

$$\begin{aligned}\left(\frac{1}{q}\right)^2 &\geq P_I(\mathcal{S}_i, W)P_S(\mathcal{S}_i, W)\\ &\geq 2^{-I(MA^*;E_i|E_W)-H(E_i|E_W,MA_{i_1},\ldots,MA_{i_j})} \qquad (3)\\ &= 2^{-H(E_i|E_W)+H(E_i|E_W,MA^*)-H(E_i|E_W,MA_{i_1},\ldots,MA_{i_j})}\\ &\geq 2^{-H(E_i|E_W)} \qquad (4)\\ &\geq 2^{-H(E_i)}\\ &\geq 2^{-\log|\mathcal{E}_i|} = \frac{1}{|\mathcal{E}_i|},\end{aligned}$$

where (3) follows from Lemma 1 and (4) follows from the deterministic algorithm *Agg*: $\mathcal{A}^{(l)} \to \mathcal{A}^*$: since it follows that $H(MA_{i_1}, \ldots, MA_{i_j}) \geq H(MA^*)$, we have $H(E_i|E_W, MA^*) - H(E_i|E_W, MA_{i_1}, \ldots, MA_{i_j}) \geq 0$. □

Lemma 3. $|\mathcal{E}_R| \geq q^{2(\omega+1)}$.

Proof. Without loss of generality, we assume that the best situation for the adversary is when he corrupts all active senders except for the only one of them since the adversary can get $|\mathcal{S}| - 1$ active senders' secret keys. Therefore, we consider the situation in this proof. For arbitrary $i \in \{1, 2, \ldots, n\}$, let $W_i := \{T_1, \ldots, T_{i-1}, T_{i+1}, \ldots, T_{i_{\omega+1}}\}$ and $\mathcal{S}_i \in \mathcal{P}(\mathcal{T})$ such that $\mathcal{S}_i - W_i = \{T_i\}$. Then, we have

$$\begin{aligned}\left(\frac{1}{q}\right)^{2(\omega+1)} &\geq \prod_{j=1}^{\omega+1} P_I(\mathcal{S}_i, W_i)P_S(\mathcal{S}_i, W_i)\\ &\geq 2^{-\sum_{i=1}^{\omega+1} H(E_i|E_{W_i})} \qquad (5)\\ &\geq 2^{-\sum_{i=1}^{\omega+1} H(E_i|E_1,\ldots,E_{i-1})}\\ &= 2^{-H(E_1,\ldots,E_{\omega+1})}\\ &\geq 2^{-H(E_v)} \qquad (6)\\ &\geq 2^{-\log|\mathcal{E}_R|} = \frac{1}{|\mathcal{E}_R|},\end{aligned}$$

where (5) follows from the same way as (4), and (6) follows from the mapping π. □

Lemma 4. $|\mathcal{A}| \geq q$.

Proof. In this lemma, for any $\mathcal{S} = \{T_{i_1}, \ldots, T_{i_j}\} \in \mathcal{P}(\mathcal{T})$, M_i and A_i denotes random variables which take values in $\mathcal{M}$ and $\mathcal{A}$, respectively, to be sent from $T_i \in \mathcal{S}$. We note that each M_i may differ from each other, and also that each M_i is independent of each other.

For arbitrary $i \in \{1, 2, \ldots, n\}$, let W and $\mathcal{S}_i = \{T_{i_1}, \ldots, T_{i_j}\}$ such that $\mathcal{S}_i - W = \{T_i\}$. Then, we have

$$\begin{aligned}
\frac{1}{q} &\geq P_I(\mathcal{S}_i, W) \\
&= 2^{-H(MA^*|E_W)} && (7) \\
&\geq 2^{-H(MA_{i_1}, \ldots, MA_{i_j}|E_W)} && (8) \\
&= 2^{-H(MA_i|E_W)} \\
&= 2^{-I(MA_i;E_v|E_W)} \\
&= 2^{-I(M_i;E_v|E_W) - I(A_i;E_v|E_W, M_i)} \\
&\geq 2^{-H(A_i)} \geq \frac{1}{|\mathcal{A}|},
\end{aligned}$$

where (7) follows from Theorem 1 and (8) follows from the deterministic algorithm *Agg*: $\mathcal{A}^{(i_j)} \to \mathcal{A}^*$. □

Lemma 5. $|\mathcal{A}^*| \geq q$.

Proof. By the assumption $|\mathcal{A}^*| \geq |\mathcal{A}|$ (or equivalently, $\gamma \geq 1$) and Lemma 4, it is clear that $|\mathcal{A}^*| \geq q$. □

As we will see in Section 4.2, the above lower bounds are all tight since our direct construction will meet all the above inequalities with equalities. Therefore, we define optimality of constructions of aggregate A-codes as follows.

Definition 4. A construction of aggregate A-codes is said to be *optimal*, if it is $(n, \omega, \epsilon, 1)$-one-time secure (i.e., $\gamma = 1$) and it meets equality in every inequality of (i)-(iv) in Theorem 2.

Remark 1. It should be noted that, in the case of $|\mathcal{S}| = 1$, $W = \emptyset$, and the algorithm *Agg* being the identity mapping, the lower bounds of aggregate A-codes in Theorem 2 are the same as those of traditional A-codes [12]. Namely, our results on aggregate A-codes are regarded as extension of those of A-codes.

4 Constructions

In this section, we propose two kinds of constructions of $(n, \omega, \epsilon, \gamma)$-one-time secure aggregate A-codes.

4.1 Simple Generic Construction

We introduce a simple generic construction of $(n, \omega, \epsilon, \gamma)$-one-time secure aggregate A-codes starting from only traditional A-codes (e.g., see [12]). First, we briefly explain the traditional A-codes as follows.[2]

A-code. We consider a scenario where there are three entities, a sender S, a receiver R and an adversary A. The A-code Θ consists of a three-tuple of algorithms (*AGen*, *Tag*, *Ver*) with three spaces, $\tilde{\mathcal{M}}$, $\tilde{\mathcal{A}}$ and $\tilde{\mathcal{E}}$, where $\tilde{\mathcal{M}}$ is a finite set of possible messages, $\tilde{\mathcal{A}}$ is a finite set of possible tags (or authenticators) and $\tilde{\mathcal{E}}$ is a finite set of possible secret-keys, respectively. *AGen* is a key generation algorithm, which takes a security parameter on input and outputs a secret-key e. *Tag* is a deterministic algorithm for generating a tag. *Tag* takes a message $m \in \tilde{\mathcal{M}}$ and a secret-key $e \in \tilde{\mathcal{E}}$ on input and outputs a tag $\alpha \in \tilde{\mathcal{A}}$, and we write $\alpha = Tag(m, e)$ for it. On receiving α, a receiver R can check the validity of it by using *Ver*. *Ver* takes a message m, a tag α and a secret-key e on input, and outputs *true* or *false*, and we write $true = Ver(m, \alpha, e)$ or $false = Ver(m, \alpha, e)$ for it. In A-codes, there are two kind of attacks: *impersonation attacks* and *substitution attacks*. Here, Θ is said to be *ϵ-secure* if each of success probabilities of these attacks is at most ϵ.

The detail of our generic construction of aggregate A-codes Π=(*KGen, Auth$_i$, Agg, Vrfy*) by using A-codes Θ=(*AGen, Tag, Ver*) is given as follows.

1. ***KGen***. For a security parameter 1^k, *KGen* outputs matching secret-keys e_i and e_v for T_i $(1 \leq i \leq n)$ and R, respectively, as follows. *KGen* calls *AGen* with input 1^k n times, and suppose its output is $(e^{(1)}, e^{(2)}, \ldots, e^{(n)})$, where $e^{(i)}$ is the i-th output by *AGen*. Then, *KGen* outputs secret-keys $e_i := e^{(i)}$, and $e_v := (e^{(1)}, \ldots, e^{(n)})$ for T_i $(1 \leq i \leq n)$ and R, respectively.
2. ***Auth$_i$***. For a message m_i which T_i wants to authenticate and a secret-key $e_i = e^{(i)}$, *Auth$_i$* calls *Tag*, and it computes a tag $\alpha^{(i)} = Tag(m_i, e^{(i)})$. Finally, *Auth$_i$* outputs $tag_i := \alpha^{(i)}$.
3. ***Agg***. For tags $(tag_{i_1}, \ldots, tag_{i_j}) = (\alpha^{(i_1)}, \ldots, \alpha^{(i_j)})$, *Agg* computes an aggregated tag *tag* by XORing all tags: $tag := \bigoplus_{k=1}^{j} \alpha^{(i_k)}$. Then, *Agg* outputs it.
4. ***Vrfy***. For $M := ((m_{i_1}, T_{i_1}), \ldots, (m_{i_j}, T_{i_j}))$, an aggregated tag *tag*, and a verification-key $e_v = (e^{(1)}, \ldots, e^{(n)})$, *Vrfy* calls *Tag* with inputting them, and suppose $\alpha^{(i_k)} = Tag(m_{i_k}, e^{(i_k)})$ for all $1 \leq k \leq j$ such that $T_{i_k} \in \mathcal{S}$. Then, *Vrfy* outputs *true* if and only if $tag = \bigoplus_{k=1}^{j} \alpha^{(i_k)}$.

The security of the above construction is shown as follows.

Theorem 3. *Given an ϵ-secure A-code Θ, then the aggregate A-code Π formed by the above construction based on Θ is $(n, \omega, \epsilon, \gamma)$-one-time secure, where $\omega = n - 1$ and $\gamma = 1$. Furthermore, memory-sizes of tags and secret-keys required in the above construction are given by*

$$|\mathcal{E}_i| = |\tilde{\mathcal{E}}|, \ |\mathcal{E}_R| = |\tilde{\mathcal{E}}|^n, \ |\mathcal{A}^*| = |\mathcal{A}| = |\tilde{\mathcal{A}}|.$$

[2] More precisely, we explain *Cartesian A-codes without splitting* in this paper.

Proof Sketch. The proof can be easily shown by the security of the underlying A-code, and the estimation of memory-sizes is straightforward. Here, we only describe the outline of the proof of $P_S \leq \epsilon$, since $P_I \leq \epsilon$ can be shown by a similar idea.

Without loss of generality, we suppose that $\mathcal{S} = \mathcal{T}$ and $W = \mathcal{T} - \{T_n\}$. The adversary can know $n-1$ secret-keys from corrupted senders and n valid pairs of messages and tags, however, he cannot know T_n's secret-key $e^{(n)}$. Thus, since the underlying A-code is ϵ-secure, success probability of substitution attacks is at most ϵ. Hence, the adversary cannot guess the aggregated tag $tag := \bigoplus_{i=1}^{n} \alpha^{(i)}$ with probability larger than ϵ. Therefore, we have $P_S \leq \epsilon$. In manner similar to this, we can prove $P_I \leq \epsilon$. Hence, we have $\max(P_I, P_S) \leq \epsilon$. □

Remark 2. This generic construction is very simple. However, even if we apply optimal constructions of A-codes in the above generic construction, we cannot obtain an optimal construction of aggregate A-codes for any ω except $\omega = n - 1$. Therefore, in the next subsection we will show that there exists a direct construction (i.e., a construction from scratch) which satisfies Definition 4 for any $\omega(< n)$.

4.2 Optimal Direct Construction

We propose a direct construction of $(n, \omega, \epsilon, \gamma)$-one-time secure aggregate A-codes. In addition, it is shown that the construction is optimal. The detail of our construction of aggregate A-codes, Π=(*KGen, Auth$_i$, Agg, Vrfy*), is given as follows.

1. ***KGen***. For a security parameter 1^k, *KGen* outputs matching secret-keys e_i and e_v for T_i $(1 \leq i \leq n)$ and R, respectively, as follows. *KGen* picks a k-bit prime power q, where $q > n$, and constructs the finite field $\mathbb{F}_q$ with q elements. We assume that the identity of each user T_i is encoded as $T_i \in \mathbb{F}_q \backslash \{0\}$. And, *KGen* chooses uniformly at random $f(x) := \sum_{i=0}^{\omega} a_i x^i$ and $g(x) := \sum_{i=0}^{\omega} b_i x^i$ over $\mathbb{F}_q$ with a variable x in which a degree of x is at most ω. *KGen* also computes $e_i := (f(T_i), g(T_i))$ $(1 \leq i \leq n)$. Then, *AGen* outputs secret-keys e_i $(1 \leq i \leq n)$ and $e_v := (f(x), g(x))$ for T_i $(1 \leq i \leq n)$ and R, respectively.
2. ***Auth$_i$***. For a message $m_i \in \mathbb{F}_q$ which T_i wants to authenticate and a secret-key e_i, *Auth$_i$* generates a tag, $tag_i := f(T_i)m_i + g(T_i)$, and outputs it.
3. ***Agg***. For tags $(tag_{i_1}, \ldots, tag_{i_j})$, *Agg* computes an aggregated tag, $tag := \sum_{k=1}^{j} tag_{i_k}$. Then, *Agg* outputs it.
4. ***Vrfy***. For $M = ((m_{i_1}, T_{i_1}), \ldots, (m_{i_j}, T_{i_j}))$, an aggregated tag tag, and a verification-key e_v, *Vrfy* outputs *true* if $tag = \sum_{k=1}^{j} f(T_{i_k})m_{i_k} + g(T_{i_k})$ holds, and otherwise outputs *false*.

The security and optimality of the above construction is stated as follows.

Theorem 4. *The resulting aggregate A-code Π by the above construction is $(n, \omega, \frac{1}{q}, 1)$-one-time secure and optimal.*

Proof Sketch. Here, we only describe the outline of the proof of $P_S \leq \frac{1}{q}$, since $P_I \leq \frac{1}{q}$ can be shown by a similar idea.

Without loss of generality, we suppose that $W = \{T_1, \ldots, T_\omega\}$, $T_n \in \mathcal{S}$, and $T_n \notin W$. To succeed in the substitution attack by an adversary who corrupts the colluding group W, the adversary will generate fraudulent messages and an fraudulent aggregated tag (M', tag') under the following conditions: the adversary can obtain ω secret-keys from corrupted senders, and $|\mathcal{S}|$ valid pairs of messages and tags, one of the pairs is generated by T_n. However, each degree of $f(x)$ and $g(x)$ with respect to x is at most ω, the adversary cannot guess at least one coefficient of $f(x)$ and $g(x)$ with probability larger than $1/q$. Therefore, we have $P_S \leq 1/q$. In a manner similar to this, we can prove that $P_I \leq 1/q$. Thus, we have $\max(P_I, P_S) \leq 1/q$.

Finally, it is straightforward to see that the construction satisfies all the equalities of lower bounds in Theorem 2. □

5 Concluding Remarks

In this paper, we studied aggregate authentication codes (aggregate A-codes) with information-theoretic security. Specifically, we first proposed a formal model and formalization of security for aggregate A-codes. We also derived tight lower bounds on memory-sizes required for aggregate A-codes. Furthermore, we presented a simple generic construction and an optimal direct construction of aggregate A-codes.

Acknowledgments

The authors would like to thank the referees for their helpful comments. The third author is supported by JSPS Research Fellowships for Young Scientists.

References

1. Bellare, M., Namprempre, C., Neven, G.: Unrestricted Aggregate Signatures. In: Arge, L., Cachin, C., Jurdziński, T., Tarlecki, A. (eds.) International Colloquium on Automata, Languages, and Programming (ICALP) 2007, LNCS, vol.4596, pp.411-422. Springer, Heidelberg (2007)
2. Boneh, D., Gentry, C., Lynn, B., Shacham, H.: Aggregate and Verifiably Encrypted Signatures from Bilinear Maps. In: Biham, E. (ed.) EUROCRYPT 2003, LNCS, vol.2656, pp.416-432. Springer, Heidelberg (2003)
3. Castro, R., Dahab, R.: Efficient Certificateless Signatures Suitable for Aggregation. In: Cryptology ePrint Archive: Report 2007/454. (2007) Available at `http://eprint.iacr.org/2007/454`.
4. Desmedt, Y., Frankel, Y., Yung, M.: Multi-receiver/Multi-sender network security: efficient authenticated multicast/feedback. In: IEEE Infocom 1992, pp.2045-2054. (1992)

5. Johansson, T.: Lower Bounds on the Probability of Deception in Authentication with Arbitration. In: IEEE Trans. on Information Theory, vol.40, no.5, pp.1573-1585. (1994)
6. Katz, J., Lindell, Y.: Aggregate Message Authentication Codes. In: Malkin, T. (ed.) Cryptographer's Track, RSA Conference (CT-RSA) 2008, LNCS, vol.4964, pp.155-169. Springer, Heidelberg (2008)
7. Lu, S., Ostrovsky, R., Sahai, A., Shacham, H., Waters, B.: Sequential Aggregate Signatures and Multisignatures Without Random Oracles. In: Vaudenay, S. (ed.) EUROCRYPT 2006, LNCS, vol.4004, pp.465-485. Springer, Heidelberg (2006)
8. Lysyanskaya, A., Micali, S., Reyzin, L., Shacham, H.: Sequential Aggregate Signatures from Trapdoor Permutations. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004, LNCS, vol.3027, pp.74-90. Springer, Heidelberg (2004)
9. Martin, K., Safavi-Naini, R.: Multisender Authentication Systems with Unconditional Security. In: International Conference on Information and Communication Security (ICICS) 1997, LNCS, vol.1334, pp.130-143. Springer, Heidelberg (1997)
10. Okamoto, T.: A Digital Multisignatures Scheme Using Bijective Public-key Cryptosystems. In ACM Trans., Computer Systems, vol.6(4), pp.432-441. (1988)
11. Rivest, R.: Unconditionally Secure Commitment and Oblivious Transfer Schemes Using Private Channels and a Trusted Initializer. manuscript. (1999) Available at `http://people.csail.mit.edu/rivest/Rivest-commitment.pdf`
12. Simmons, G. J.: Authentication Theory/Coding Theory. In: Blakley, G.R., Chaum, D. (eds.) CRYPTO '84, LNCS, vol.196, pp.411-431, Springer, Heidelberg, (1885)