Anonymous Broadcast Authentication with Logarithmic-order Ciphertexts from DLP or LWE

Yoshinori Aono[1,2] and Junji Shikata[3,1]

[1] Institute of Advanced Sciences, Yokohama National University 79-5 Tokiwadai, Hodogaya-ku, Yokohama 240-8501, Japan.
[2] NICT, 4-2-1, Nukui-Kitamachi, Koganei, Tokyo, Japan.
[3] Graduate School of Environment and Information Sciences, Yokohama National University, 79-7 Tokiwadai, Hodogaya-ku, Yokohama 240-8501, Japan

Abstract. Herein, we propose an anonymous broadcast authentication (ABA) scheme to simultaneously control 10^9 devices practically. We find a barrier to construct an ABA working with a larger number of devices. In a nutshell, there is a trilemma between (i) security, (ii) ciphertext length, and (iii) freedom in the target devices selection. For practical use, we propose ABAs with a ciphertext size of $O(\log N)$ where N is the number of target devices while we impose a certain restriction on (iii). We provide an ABA template and instantiate it into specific schemes from the discrete logarithm problem (DLP) or the learning with errors (LWE) problem.

Keywords: Anonymous broadcast authentication · IoT Network · Discrete logarithm problem · Learning with errors problem

1 Introduction

The ABA [1] is a one-way communication between a central server and multiple resource-limited devices. The server broadcasts a command to control a subset of devices. The following conditions (1)(2) are the minimum desired specifications for correctness. (1) A message from the server includes information on the IDs of the target devices and control commands. Each device that receives the message either executes the command if the device is included in the target devices or does nothing if the device is not included in the target devices. (2) The received message has integrity and authenticity.

Also, it should satisfy two additional security notions (3)(4). (3) Unforgeability: In a situation where secret information of some devices are leaked, an entity with information cannot forge a legitimate ciphertext. (4) Anonymity: Each device can detect whether or not it is a target, but cannot determine whether another device is a target.

An application that we envision is sending emergency signals to reboot or shut down malware-infected devices. Thus, assume that the space of commands is small (a few bits) to send reboot, shutdown, or other optional flags. We expect the number of devices to be about $10^6 - 10^9$ to control all the devices within a wireless area (several square kilometers) simultaneously in the 5G IoT or the network beyond it. The entire process of command generation in a central server, communication, and authentication in target devices must also be completed within a few seconds for a fast response to an emergency.

A barrier on the length of ciphertexts has been known. The atomic model of private broadcasting encryption (pBE) by Kiaias-Samari [2] and its adaptation to ABA by Watanabe et al. [3] provide useful tool for discussing the problem. In addition to the above application, ABA is technically interesting since it is an authentication-oriented analog of anonymous BE (ANOBE). They proved that if a protocol has anonymity the ciphertext length must be $\Omega(n)$, where n is the number of target or joined devices; it depends on the security requirement of anonymity. Aside from the limitation on the atomic model, a similar bound can be considered from Shannon's coding theorem because the condition (1) requires that the amount of information contained in the ciphertext should exceed N under the assumption that the target device set is randomly selected from a family of any subset of N devices.

These observations deduce the following trilemma: In ABA, the three conditions of (i) security (anonymity), (ii) ciphertext length and (iii) freedom in the target devices selection are not simultaneously satisfied. For practical use, we propose ABAs with a ciphertext size of $O(\log N)$ while we impose a certain restriction on (iii). Concretely, our ABA protocol has the device IDs represented by a vector $(\mathsf{id}_1, \ldots, \mathsf{id}_K)$ where $\mathsf{id}_j \in [N_j] := \{1, 2, \ldots, N_j\}$ and ciphertext length $O(\sum N_j)$. It can control $\prod N_j$ devices, which is an exponential number to the ciphertext length.

We first constructed an ABA template and provided instantiations from the DLP or the LWE problem. The command ciphertext for controlling 10^9 devices with a 128-bit security had lengths of 10 KBytes and 1MBytes in the elliptic curve DLP and lattice situations, respectively.

1.1 Related Work

Atomic model: This model assumes that the server broadcasts a sequence $\mathsf{ct}_1, \ldots, \mathsf{ct}_\ell$ of the ciphertexts of the control command. Each device j then tries to decrypt each ct_i using its secret key dk_j.

The lengths of the command ciphertext in this model is well-studied in the private key broadcast encryption (prBE) [2] and ABA [3]. It has been shown that $\ell \geq N$ if an ABA controlling N devices has anonymity. If it has the weak-anonymity instead of the anonymity the lower bound can be relaxed to $\ell \geq |\mathcal{S}|$ where $\mathcal{S} \subset [N]$ is the set of target devices. The bit lengths of total ciphertexts are bounded by $\Omega(N \cdot \lambda)$ and $\Omega(|\mathcal{S}| \cdot \lambda)$ where λ is the security parameter. In both cases, concrete constructions achieving the bounds are given by [1, 3].

Broadcast encryption: ABA is considerably similar to broadcast encryption (BE). At the formal definition level, the notion of ABA is equivalent to pBE in [4, Def. 3.1]. Conversely, security notions are slightly different. Their gap is mainly from the application.

The anonymity notion in the BE framework was proposed by Barth et al. [5] They assumed that the condition $|\mathcal{S}_0| = |\mathcal{S}_1|$, corresponds to the weak anonymity in the ABA framework. They also assumed that the adversary can get all the secret keys in $\mathcal{S}_0 \cap \mathcal{S}_1$. This is in contrast to the setting in the ABA anonymity game, in which the adversary can select id of which one wants the secret (verification) keys. After the work, an efficient scheme is given by Benoît et al. [6] with the notion of ANOBE. Fazio et al. [7] gives a log-order ciphertext scheme within the public key BE.

Following the existing works, we mention two issues when we import the techniques in BE to our ABA. First, constructing a practical scheme with short, i.e., $o(N)$ [bit] ciphertext with keeping reasonable anonymity has been considered as one of the challenging problems in BE and its variants. It is not trivial to import an existing scheme and give a proof of ABA anonymity. On the other hand, a transformation technique to add the unforgeability can be used as explained in the following paragraph.

Interpreting the authentication result as the transmission of an 1 bit message, the framework of ABA can be considered as a prBE with additional functionalities. Our scheme in this paper can be considered as a new result of short ciphertext prBE, besides the context of ABA.

Transformation to add unforgeability: A similarity between the unforgeability of ABA and the CCA1 security of prBE should be considered. Although their goals are different (i.e., forge and distinguish), the abilities of the adversaries in security games are similar. In the CCA1 game of pkBE, an adversary can have a polynomial number of accesses to corrupt, encrypt, and decrypt oracles. However, an adversary can access the corrupt and encryption oracles in the t-unforgeability game of ABA, where access to the corrupt oracle is limited within t times.

Therefore, transformation techniques from a weak-security BE to strong-security BE can be used to construct an ABA with unforgeability. The simplest form of this conversion should be the simple addition of a signature to the broadcasting ciphertext as in [8].

Rough estimation of ciphertext size: Consider a situation in which a standard encryption and a signature schemes is used as base gadgets to construct an atomic type ABA. For example, a standard ElGamal encryption requires a kilobyte ciphertext, while a standard (resp. structured) lattice-based encryption needs tens of kilobytes (resp. a half of kilobyte) length ciphertexts; FrodoKEM [9] and FALCON [10] provide good examples of sizes after optimization. This implies that a system for controlling $N = 10^6$ devices requires ciphertexts for sending a ciphertext is presented in gigabytes, which is too large to rapidly process on low-resource devices. Thus, an ABA with short ciphertexts is necessary to control millions of devices.

1.2 Our Contributions

Design rationale: We first explain our construction strategy from the viewpoint of lower bound arguments. Let us organize the considered conditions and the linear lower bound of the ciphertext length. We used At and An to represent ABA with an atomic model and anonymity, respectively. LB represents ABA with ciphertexts longer than N or $|\mathcal{S}|$ in average over the selection of target sets and messages. Then, the result of Kobayashi et al. [3] can then be roughly described as $[At \text{ AND } An] \Rightarrow LB$.

In addition, we denote F as the freedom in the choice of target devices, i.e., any $\mathcal{S} \subset [N]$ can be selected as a target device, and assume that it is randomly chosen from $2^{[N]}$. According to Shannon's coding theorem, the ciphertext must be longer than N bits in average because the broadcasting ciphertext entropy exceeds N bits. We remark that talking a variable ciphertext length depending on $\mathcal{S}$ is a possible way to reduce the communication cost in many transmissions because the distribution of $\mathcal{S}$ should be low entropy in many practical situations. However, the length of ciphertext should leak information on $\mathcal{S}$, and thus we conclude the variable ciphertext length is not a better strategy to keep the security.

Thus, we have $F \Rightarrow LB$ and the following relation

$$\neg LB \Rightarrow \neg F \text{ AND } [\neg At \text{ OR } \neg An].$$

An ABA with short ciphertexts must restrict conditions among F, At and An. We emphasize our construction is in an edge of the conditions. It satisfies $\neg F$, $\neg At$, and a nearly anonymity.

We also explain why we restrict the strength of anonymity by introducing a new notion of anonymity. The single anonymity (Appendix A) which we denote SA, is a notion about information leakage on $\mathsf{id}' \overset{?}{\in} \mathcal{S}$ from other patterns $\{\mathsf{id} \overset{?}{\in} \mathcal{S}\}$. This notion is used to discuss the situation where the freedom in the choice of target devices are limited. We proved that $\neg F \Rightarrow \neg SA$ that derives $SA \Rightarrow F \Rightarrow LB$. Thus, any short ciphertext ABA inherently lacks the single anonymity, and this is the reason that we do not investigate the anonymity for our ABA in Section 5.2.

Construction of ABA with logarithmic-order We first considered a Vernam-styled multirecipient encryption (MRE) as a fundamental gadget with information-theoretic security. However, it cannot provide security if the server sends ciphertexts with the same secret key many times. Thus, a Vernal-styled ABA might be useless in practice even though it has very high performance. To address this problem, we transformed MRE to a computationally secure ABA with a template function f_{prm} using the technique of Kurosawa et al. [11].

From the template, we instantiated a discrete logarithm version of the protocol. The former is not secure against a large scale quantum computer and is essentially similar with concatenation of the naïve multirecipient encryption in [12, Sect. 1.3]. We also instantiated an LWE version throughout the template.

The above template construction was within the atomic model that sends $M \geq N$ ciphertexts to N devices. To control an exponential number of devices by a short ciphertext, we consider a concatenation of the basic template ABAs. However, the simple concatenated ABA does not have unforgeability and anonymity. Herein, we propose a modification using the idea of Agrawal et al.'s inner-product encryption [13].

Each device is indicated by a vector $(i_1, \ldots, i_K)$ where $i_j \in [N_j] := \{1, 2, \ldots, N_j\}$ in the concatenated ABAs. The target set is defined by a sequence of sets $S_j \subset [N_j]$ and a device is a target if $i_j \in S_j$ for all j. A trade-off between the ciphertext length and the flexibility of target sets can be considered by changing N_j. For instance, setting all N_j equivalent, it derives an ABA controlling N devices by $O(N^{1/K})$ [bit] ciphertext. In the extreme case, taking $N_j = 2$ for all j, it provides an ABA to control N devices by $O(\log N)$ [bit] ciphertext.

Data sizes Table 1 and 2 show the size estimation of a verification key and a command ciphertext in our scheme to control 2^{20} to 2^{30} devices. The first table gives the classical setting whose security is based on the standard and elliptic curve DLP with parameter that are claimed 112 bit security.

Table 2 and 3 gives the parameter and sizes in the post-quantum setting whose security is based on the LWE problem that are claimed 128 security. The details of parameters and security estimation will be described in Section 6.2.

Table 1. Sizes of a verification key $(2KM(\log_2 q)/8 + \mathsf{pksize}$ [Byte]) and command ciphertext $((1 + 4KM)(\log_2 q)/8 + \mathsf{sigsize}$ [Byte]). Here, $M = 3$ is the dimension of base vector, K is number of concatenated base ABAs, which makes possible to control 2^K devices, and q is the bitsize of a modulus to define finite fields. We assume $\lfloor \log_2 q \rfloor = 2048$ in the DLP construction, and assume it spends 256 bits by a compressed representation in Curve25519 in the ECDLP setting. pksize and $\mathsf{sigsize}$ are the sizes of public key and signature in a strongly and existentially unforgeable signature. We assume they are $(\log_2 q)/8$[bit] and $2 \cdot (\log_2 q)/8$[bit] following the DSA scheme.

		Size(vk) [Byte]	Size(cmd) [Byte]
Finite field	$K = 20$	30976	62208
(112 bit security)	$K = 30$	46336	92928
Elliptic curve	$K = 20$	3872	7776
(112 bit security)	$K = 30$	5792	11616

1.3 Paper Organization

Section 2 is the preliminary section. We provide background definitions, notations, and theorems. In particular, computational problems on discrete logarithms and lattices that are bases of the security of ABA are introduced, together with notion and security definitions of ABA. In Section 3, we define Vernam-styled multirecipient encryption (MRE) which allows broadcasting an encrypted message only for target devices with information-theoretic security. In Section 4,

Table 2. Sizes of a verification key $((4Kn)\lfloor \log_2 q \rfloor/8 + \mathsf{pksize})$ and command ciphertext $((1+8Kn)\lfloor \log_2 q \rfloor/8 + \mathsf{sigsize})$ in our lattice based ABA. pksize and sigsize are the size of public key and signature of a strongly and existentially unforgeable signature. We assume that pksize=897 [Byte] and sigsize=666 [Byte] from 128 bit security FALCON signature [10]. K sets the number of participant devices 2^K, $\sigma = 3.0$ for all settings. L and Q are the variety of message and buffer to prevent the overflow; see, Section 6.2 for detail.

Security Level	(K, L)	n	(q, Q)	Size(vk) [Byte]	Size(cmd) [Byte]
128 bit	(20,4)	926	(68588467, 428678)	250917	500710
	(20,256)	1164	(4921551113, 480621)	385017	768911
	(30,4)	961	(128364259, 534852)	390102	779080
	(30,256)	1119	(9176392691, 597422)	612387	1223651

Table 3. Sizes of a verification key $((4Kn)\lfloor \log_2 q \rfloor/8 + \mathsf{pksize})$ and command ciphertext $((1+8Kn)\lfloor \log_2 q \rfloor/8 + \mathsf{sigsize})$ in our lattice based ABA. pksize and sigsize are the size of public key and signature of a strongly and existentially unforgeable signature. We assume that pksize=1793 [Byte] and sigsize=1280 [Byte] from 256-bit security FALCON signature [10]. Other settings are the same as Table 2.

Security Level	(K, L)	n	(q, Q)	Size(vk) [Bytes]	Size(cmd) [Bytes]
192 bits	(20,4)	1373	(83518147, 521989)	372503	742704
	(20,256)	1715	(5973894821, 583389)	567743	1133185
	(30,4)	1423	(156201391, 650840)	599453	1196604
	(30,256)	1765	(11133577901, 724843)	901943	1801585
256 bits	(20,4)	1799	(95600731, 597505)	487523	972744
	(20,256)	2238	(6824259821, 666432)	740333	1478365
	(30,4)	1863	(178726489, 744694)	784253	1566204
	(30,256)	2302	(12714961717, 827797)	1175813	2349325

we convert the MRE to the computationally secure ABA with a template function f_{prm}. From the template, we instantiate the (standard and elliptic curve) discrete logarithm problem and learning with errors (LWE) versions. In Section 5, we propose an ABA of short command ciphertext by reducing the freedom in the choice of target devices. Section 6 provides concrete protocols based on discrete logarithms and lattices, and provides parameter sets and sizes of command ciphertexts for practical usage. Finally, Section 7 gives concluding remarks and future discussions.

2 Preliminaries

$\mathbb{Z}$ and $\mathbb{N}$ are the set of integers and natural numbers. For $N \in \mathbb{N}$, denote the set $[N] := \{1, \ldots, N\}$. Define $\mathbb{Z}_q := \{0, 1, \ldots, q-1\}$ and q is assumed to be an odd prime. $\mathbb{Z}_q^\times := \mathbb{Z}_q \setminus \{0\}$. For a finite set A, let the notation $a \overset{\$}{\leftarrow} A$ be the uniform sampling. Bold letters such as $\boldsymbol{c}$ represent a row vector. The transpose notation $\boldsymbol{c}^T$ represents a column vector, We use $\boldsymbol{u}_i$ to denote the i-th unit vector $(0,\ldots,1,\ldots,0)$ whereas the dimension is omitted if it is clear from the context. For vectors and matrices, the notation $||$ denotes the concatenation.

2.1 Computational Problems

We introduce the (EC)DLP and LWE, used as security bases of our ABA.

The finite field version of the DLP is to find an integer z that satisfies $g^z \equiv a \pmod{q}$ from a given tuple (g, a, p). Here, q is a prime that defines the finite field $\mathbb{Z}_q$. g is a generator, that is, any integer $h \in \{1, \ldots, q-1\}$ can be written as $g^i \bmod q$ by using some $i \in \mathbb{Z}$. a is an integer between 2 and $q-1$. The DDH assumption is to distinguish (g^a, g^b, g^{ab}) and (g^a, g^b, g^r) where a, b, r are random from $\mathbb{Z}_q$. The recommended parameter size is 2048 bits that achieves 112 bit security [14, Sect. 5.5.1.1].

Variants of the DLP using other groups are considered. In particular, elliptic curve instantiations are the most successful results that can reduce the communication cost with keeping security. The Curve25519 used in the Ed25519 signature has 256 bit length public key and 512 bit length signatures keeping a 112 bit security by using a compressed representation of a point on the curve.

The LWE problem [15] is a fundamental toolkit for constructing lattice based schemes. For a dimension parameter n, a modulo q, and an error distribution χ, the decision LWE is defined by the problem to distinguish the samples $\{(\boldsymbol{a}_i, \boldsymbol{a}_i\boldsymbol{s}^T + e_i)\}_{i=1,\ldots,m}$ and $\{(\boldsymbol{a}_i, u_i)\}_{i=1,\ldots,m}$ where $\boldsymbol{s}^T \in \mathbb{Z}_q^n$ is a random secret vector fixed the all samples. $\boldsymbol{a}_i$, e_i, u_i are random vectors from $\mathbb{Z}_q^n$, random errors from χ, and random elements from $\mathbb{Z}_q$ respectively. χ is typically the discrete Gaussian distribution $D_{\mathbb{Z},\sigma}$ whose density function defined over $\mathbb{Z}$ is $\Pr[X = x] \propto \exp(-x^2/2\sigma^2)$. The goal of the search version of LWE is to recover $\boldsymbol{s}$ from legitimate samples $\{\boldsymbol{a}_i, \boldsymbol{a}_i \cdot \boldsymbol{s}^T + e_i\}_{i=1,\ldots,m}$. The polynomial time equivalence between decision and search is known [15]. We set the lattice parameter using Albrecht et al.'s lattice estimator [16] as of May 2022.

2.2 Anonymous Broadcast Authentication (ABA)

The notion of ABA is stated by Watanabe et al. [1].

Definition 1. *An ABA is formally defined by the tuple of four functions $\mathit{\Pi} = (\mathsf{Setup}, \mathsf{Join}, \mathsf{Auth}, \mathsf{Vrfy})$.*

- $\mathsf{Setup}(1^\lambda, N, \mathcal{D}) \to \mathsf{ak}$*: An algorithm that outputs the authorization key* ak *from its inputs.* 1^λ *is a security parameter,* N *is the maximum number of joined devices, and* $\mathcal{D}$ *is a family of sets* $\mathcal{S} \subset [N]$ *allowed to use as a set of the target device.*
- $\mathsf{Join}(\mathsf{ak}, \mathsf{id}) \to \mathsf{vk}_{\mathsf{id}}$*: An algorithm that outputs a verification key* $\mathsf{vk}_{\mathsf{id}}$ *embedded to the device* id*.*

- $\mathsf{Auth}(\mathsf{ak}, \mathsf{m}, \mathcal{S}) \to \mathsf{cmd}_{\mathcal{S}}$*: It outputs a command ciphertext that encrypts the information of the message* m *and the set* $\mathcal{S}$ *of the target devices.*
- $\mathsf{Vrfy}(\mathsf{vk}_i, \mathsf{cmd}_{\mathcal{S}}) \to \mathsf{m}/\mathsf{reject}$*: It verifies the command ciphertext* $\mathsf{cmd}_{\mathcal{S}}$ *using the verification key* $\mathsf{vk}_{\mathsf{id}}$*. It returns the message or* reject *if it was accepted or rejected, respectively.*

The abovementioned algorithms, except for Vrfy are assumed to be probabilistic polynomials. The family $\mathcal{D}$ is typically set as $2^{[N]}$ in several early works whereas we restrict the freedom in the choice of a subset in $[N]$ to construct a short ciphertext ABA.

We introduce the correctness, unforgeability, and anonymity notions of ABA. They are essentially same as in the original work of Watanabe et al. [1] whereas we explicitly mention t the number of corrupted devices.

Definition 2. *We say an ABA Π has the correctness if for any fixed* $(1^\lambda, N, \mathcal{D})$, ak *that are allowed to input,* $\mathcal{S} \in \mathcal{D}$*, and any* $\mathsf{m}, \mathsf{id} \in [N]$,

$$\Pr[\mathsf{Vrfy}(\mathsf{Join}(\mathsf{ak}, \mathsf{id}), \mathsf{Auth}(\mathsf{ak}, \mathsf{m}, \mathcal{S})) \to \mathsf{m}] = 1 - \mathsf{negl}(\lambda) \text{ if } \mathsf{id} \in \mathcal{S}, \textit{ and}$$

$$\Pr[\mathsf{Vrfy}(\mathsf{Join}(\mathsf{ak}, \mathsf{id}), \mathsf{Auth}(\mathsf{ak}, \mathsf{m}, \mathcal{S})) \to \mathsf{reject}] = 1 - \mathsf{negl}(\lambda) \text{ if } \mathsf{id} \notin \mathcal{S}$$

hold. Here, the probability is over the random coins in Join *and* Auth *(and possibly* Vrfy*).*

Below are the game-based formal definitions of unforgeability and anonymity within the situation where the receiver devices are colluded and can share their verification keys.

Definition 3. *(t-unforgeability [1]) Consider the game between a challenger* C *and an adversary* A*.*

0: C *and* A *share* $(1^\lambda, N, \mathcal{D})$ *and* C *runs* $\mathsf{Setup}(1^\lambda, N, \mathcal{D}) \to \mathsf{ak}$*. Let* $M_a = M_v = \phi$ *be the messages used in the authentication and verification queries. Also, let* $D \subset [N]$ *and* $W \subset D$ *be the set of considered devices during the game, and the set of colluded devices.* $\mathit{flag} \in \{0, 1\}$ *is a variable that indicates whether the adversary gets the success forging.*

1: (Key generation) A *selects a set of considered devices* $D \subset [N]$ *and send it to* C*.*

2: (Collusion query) A *selects* $\mathsf{id} \in D$ *and send it to* C*.* C *runs* $\mathsf{Join}(\mathsf{ak}, \mathsf{id}) \to \mathsf{vk}_{\mathsf{id}}$*. Add* id *to* W *and send back* $\mathsf{vk}_{\mathsf{id}}$ *to* A*.* A *can repeat this step until the number of colluded devices is less than* t*.*

3: (Authentication query) A *sends* $(\mathsf{m}, \mathcal{S})$ *to* C *where the selection is limited within* $\mathcal{S} \subset D$ *and* $\mathsf{m} \notin M_v$*. Then,* C *runs* $\mathsf{Auth}(\mathsf{ak}, \mathsf{m}, \mathcal{S}) \to \mathsf{cmd}_{\mathcal{S}}$ *and returns the command ciphertext.*

4: (Verification query) A *generates a set* $(\mathsf{m}, \mathsf{id}, \mathsf{cmd}_{\mathcal{S}})$ *and send them to* C*.* C *runs* $\mathsf{Vrfy}(\mathsf{vk}_{\mathsf{id}}, \mathsf{cmd}_{\mathcal{S}})$ *and returns the output to* A*. If* $\mathsf{Vrfy}(\mathsf{vk}_{\mathsf{id}}, \mathsf{cmd}_{\mathcal{S}}) = \mathsf{m}$*,* $\mathsf{id} \notin W$ *and* $\mathsf{m} \notin M_a$*, set* $\mathit{flag} = 1$ *else set* $\mathit{flag} = 0$*. Add* m *to* M_v*.*

After repeating Steps 3 and 4, if there exists a verification trial such that $\mathit{flag} = 1$*, we define the output of the experiment* $\mathsf{Exp}^{\mathsf{CMA}}_{\Pi,\mathsf{A}}(\lambda, N, \ell)$ *is* 1*, and otherwise it is* 0*. The advantage of* A *on the protocol* Π *is*

$$\mathsf{Adv}^{\mathsf{CMA}}_{\Pi,\mathsf{A}}(\lambda, N, \ell) := \Pr[\mathsf{Exp}^{\mathsf{CMA}}_{\Pi,\mathsf{A}}(\lambda, N, \ell) \to 1].$$

We say the ABA protocol Π *has* t*-unforgeability if the advantage is a negligible function of* λ*.*

The above formal definition can be interpreted as follows. Suppose t devices are taken over and colluded. Under a situation where an attacker collects secret information in these devices, it cannot forge a legitimate command ciphertext that an uncolluded device accepts. We will construct our unforgeable ABA from a base ABA by adding a signature.

We deal with the following passive attack rather than the above active attack.

Definition 4. *[1] (t-anonymity) Consider the game between a challenger* C *and an adversary* A*. As the definition of unforgeability,* t *indicates the number of colluded devices.*

0: C *and* A *share* $(1^\lambda, N, \mathcal{D})$ *and* C *runs* $\mathsf{Setup}(1^\lambda, N, \mathcal{D}) \to \mathsf{ak}$*. Let* $M_a = \phi$ *be the set of the command used in the authentication. Also, let* $D \subset [N]$ *and* $W \subset D$ *be the set of considered devices during the game, and the set of colluded devices.*

1,2: The same as the Steps 1,2 in the unforgeability game (Definition 3)

3: (Authentication query) A *selects a pair* $(\mathsf{m}, \mathcal{S}), \mathcal{S} \subset D, \mathsf{m} \notin M_a$ *and send it to* C*. Then,* C *runs* $\mathsf{Auth}(\mathsf{ak}, \mathsf{m}, \mathcal{S}) \to \mathsf{cmd}_{\mathcal{S}}$ *and return the output and adds* m *to* M_a*.*

4: (Challenge query) A *selects a command* $\mathsf{m} \notin M_a$ *and two sets of devices* $\mathcal{S}_0, \mathcal{S}_1$ *and send them to* C*.* C *runs* $\mathsf{Auth}(\mathsf{ak}, \mathsf{m}, \mathcal{S}_b) \to \mathsf{cmd}_{\mathcal{S}_b}$ *where* $b \in \{0, 1\}$ *is a random bit. Return the ciphertext to* A *and the adversary guesses* b' *for the random bit.*

We define the output of the game is 1 *if* $b = b'$*, i.e., the adversary succeeds in guessing, and the output is* 0 *if otherwise. The advantage is*

$$\mathsf{Adv}^{\mathsf{ANO}}_{\Pi,\mathsf{A}}(\lambda, N, \ell) := \left| 2 \Pr\left[\mathsf{Exp}^{\mathsf{ANO}}_{\Pi,\mathsf{A}}(\lambda, N, \ell) \right] - 1 \right|.$$

In Step 4, the considered sets $\mathcal{S}_0$ *and* $\mathcal{S}_1$ *must satisfy*

$$(\mathcal{S}_0 \triangle \mathcal{S}_1) \cap W = \phi \tag{1}$$

to prevent a trivial distinguishing; if the set $\mathcal{S}_d := (\mathcal{S}_0 \triangle \mathcal{S}_1) \cap W \neq \phi$*,* A *can check whether some* $\mathsf{id} \in \mathcal{S}_d$ *is in* $\mathcal{S}_0$ *or not via the decryption oracle.*

We pointed out that the condition (1) does not hide the size of sets. The notion of weak anonymity is defined by adding the condition $|\mathcal{S}_0| = |\mathcal{S}_1|$ besides (1) in Step 4. Also, the outsider anonymity is defined by replacing (1) with $(\mathcal{S}_0 \cup \mathcal{S}_1) \cap W = \phi$ in Step 4. It is slightly weaker than the weak anonymity, though it has no restriction on the size of sets [1].

3 Vernam-Styled Multirecipient Encryption with Information-theoretic Security

As a base gadget to construct our ABA, we introduce a simple multirecipient secret key encryption. It is a noninteractive communication protocol from a central server to N participant devices. The server packs a set of messages into one ciphertext and broadcasts it to the devices. Each device decrypts the ciphertext with its key. It has information-theoretic security on messages; that is, each device i can recover the i-th message m_i whereas it can gain no information on the messages m_j $(j \neq i)$ to the other devices. We give the actual Vernam-styled construction in the following subsection.

Definition 5. *A multirecipient encryption (MRE) is formally defined by a tuple of three functions* $\mathsf{MRE} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$.

- $\mathsf{MRE.KeyGen}(N, \mathsf{pp}) \to (\mathsf{ek}, \mathsf{dk}_1, \ldots, \mathsf{dk}_N)$*: It outputs output an encryption key* ek *and a sequence of decryption keys* $\{\mathsf{dk}_i\}_{i\in[N]}$ *for the number N of participant devices.*
- $\mathsf{MRE.Enc}(\mathsf{ek}, \{m_i\}_{i\in S}) \to \mathsf{ct}$*: S is the set of target devices to which messages are sent. m_i is a message send to i-th device and the server can take different m_i for each i. An encryption algorithm outputs* ct *to be broadcasted.*
- $\mathsf{MRE.Dec}(\mathsf{dk}_i, \mathsf{ct}) \to m_i'$*: A decryption algorithm that recovers a message in i-th device from* ct *by using one's secret key* dk_i*.*

3.1 Construction

The public parameter $\mathsf{pp} = (M, q)$ is the pair of a vector dimension and a prime modulus. With these parameters, the decryption keys are randomly generated independent column vectors $\mathsf{dk}_i \in Z_q^M$. The encryption key is the set $\mathsf{ek} = \{\mathsf{dk}_1, \ldots, \mathsf{dk}_N\}$. Each participant device i has pp and dk_i. For a set of target devices $\mathcal{S} \subset [N]$ and a set of messages $\{m_i\}_{i\in S}$ where $m_i \in Z_q^\times$, the ciphertext ct is a randomly chosen vector in Z_q^M that satisfies $\mathsf{ct} \cdot \mathsf{dk}_i^T \equiv m_i \pmod{q}$ for all $i \in S$. The decryption at device i is the computation of inner-product $\mathsf{ct} \cdot \mathsf{dk}_i^T \pmod{q}$. Thus, the correctness is immediate.

Since each decryption key and ciphertext are elements of Z_q^M, the sizes are $M \log_2 q$ bits, and the size of the encryption key saved in the central server is $NM \log_2 q$ bits.

This construction can be regarded as a generalization of the concatenation of Vernam cipher since under the situation where $N = M$ and all dk_i's are a multiple of i-th unit vector $\boldsymbol{u}_i^T$, the ciphertext $\mathsf{ct} = (c_1, \ldots, c_N)$ is the concatenation of c_i by which i-th device can decrypt. We note the reason for setting dk_i^T independent vectors instead of $k_i \cdot \boldsymbol{u}_i^T$ where k_is are multiples. Consider a chosen plaintext attack that an attacker can obtain a pair (m_i, ct) for an index i. Then, the decryption key can be easily found by a simple division where $\mathsf{dk}_i^T = k_i \cdot \boldsymbol{u}_i^T$. However, in the case where dk_i's are independent, the information-theoretic security can be ensured using vectors until M pairs of (m_i, ct) are obtained by the attacker. Using the M pairs, one can recover all the dk_i^T's via the solution of simultaneous equations.

3.2 Security of MRE

The information-theoretic security of the above vector-based MRE can be shown as follows. Suppose the situation where the devices $1, 2, \ldots, t$ are colluded, and an attacker wants to recover the message m_{t+1} of the device $t+1$ from ct using the leaked keys $\mathsf{dk}_1^T, \ldots, \mathsf{dk}_t^T$. In this case, the attacker can know only the fact that dk_{t+1} is independent to $\mathsf{dk}_1^T, \ldots, \mathsf{dk}_t^T$.

Suppose that the attacker guess a vector $\boldsymbol{v}^T$ and $m'_{t+1} = \mathsf{ct} \cdot \boldsymbol{v}^T$ as candidates of dk_{t+1}^T and m_{t+1}, respectively. Then, all the vectors $\boldsymbol{v}^T, 2\boldsymbol{v}^T, 3\boldsymbol{v}^T, \ldots, (q-1)\boldsymbol{v}^T \bmod q$ can also be candidates for the secret keys with equal possibility. Thus, $m'_{t+1}, 2m'_{t+1}, \ldots, (q-1)m'_{t+1}$, which are equal to the set $\mathbb{Z}_q^\times = \{1, 2, \ldots, q-1\}$, are also candidates of the message with equal possibility. It means that the information amount of attackers from ct and colluded secret keys is zero. A similar argument can prove the impossibility forging ct that embeds a message to the device $t+1$.

Although information-theoretic security in one-time broadcasting is guaranteed, any security under chosen-plaintext attacks and key reusing situations have never been proven. Specifically, suppose the situation where the attacker can obtain ct corresponding to any chosen $\{m_i\}_{i \in S}$ and any S with fixed decryption keys. The attacker can recover all dk_i^Ts by solving linear simultaneous equations if one has a sufficient number of pairs of messages and ciphertexts. Thus, a naïve construction of ABA from the above MRE might be useless. We transform it to a computationally secure ABA in the next section.

4 Template Construction of Base ABA

We transform the above information-theoretic MRE to ABA by adding a repeatable property. The main differences over the base MRE are: (1) it broadcasts the same message to a selected subset of participant devices, and (2) its security base is the hardness of a computational problem. First, we give a template construction that includes protocols based on discrete logarithms and lattices. Then, we discuss its anonymity and unforgeability.

4.1 A Template

We give a template of our base ABA using a function $f_{\mathsf{prm}}(\boldsymbol{c}^T)$ defined over an r-dimensional column vector with a parameter prm. We assume the function has a somewhat homomorphic property as follows. For scalars a, b and vectors $\boldsymbol{x}, \boldsymbol{y}$, $f_{\mathsf{prm}}(a\boldsymbol{x}^T + b\boldsymbol{y}^T) = a \circ f_{\mathsf{prm}}(\boldsymbol{x}^T) \otimes b \circ f_{\mathsf{prm}}(\boldsymbol{y}^T)$ holds with operations $(\circ, \otimes)$ to compute a linear combination of vectors $f_{\mathsf{prm}}(\sum_{i=1}^M v_i \boldsymbol{y}_i^T)$ in verification. We also assume that an inverse $f_{\mathsf{prm}}(\boldsymbol{x}^T)^{-1}$ of $f_{\mathsf{prm}}(\boldsymbol{x}^T)$ that satisfies $f_{\mathsf{prm}}(\boldsymbol{x}^T) \otimes f_{\mathsf{prm}}(\boldsymbol{x}^T)^{-1} = I$ (an unit) is easily computable.

For instance, our discrete logarithm construction assumes $r = 1$, $\mathsf{prm} = g$, a generator of a finite field and define $f_{\mathsf{prm}}(x) = g^x$ for $x \in \mathbb{Z}_q$. The homomorphic property holds with $a \circ h = h^a$ and $a \otimes b = a \cdot b$. Concretely,

$$f_g(ax + by) = g^{ax+by} = g^{ax} \otimes g^{by} = a \circ f_g(x) \otimes b \circ f_g(y)$$

holds. Also, the construction from the elliptic curve assumes $r = 1$ and $\mathsf{prm} = B$, where B is a base point with known order N. For $x \in [N-1]$, define $f_{\mathsf{prm}}(x) = xB$. Similar to the finite field situations, the operations are defined by $a \circ P = aP$ and $a \otimes b = a \cdot b \bmod N$.

On the other hand, our lattice-based construction uses $\mathsf{prm} = \boldsymbol{p} \in \mathbb{Z}_q^r$. and $f_{\boldsymbol{p}}(\boldsymbol{x}^T) = \boldsymbol{p}\boldsymbol{x}^T$. The homomorphic property holds with defining $a \circ \boldsymbol{x} = a\boldsymbol{x} \bmod q$ and $\boldsymbol{x} \otimes \boldsymbol{y} = \boldsymbol{x} + \boldsymbol{y} \bmod q$ for an integer a and vectors $\boldsymbol{x}, \boldsymbol{y}$. The proof is immediate. Remark that the above property holds for any vector $\boldsymbol{p}$ though we use a small $\boldsymbol{p}$ of which each coordinate is from a discrete Gaussian.

Definition 6. *(*MRE*-based ABA template) Assume the function* $f_{\mathsf{prm}}(\cdot)$ *has the above homomorphic property. A template of our ABA is defined as follows. Assume that* $\mathsf{pp} = (M, q)$ *in* MRE *is fixed from the security parameter* λ*.*

- $\mathsf{Setup}(1^\lambda, N, \mathcal{D}) \to \mathsf{ak}$*: Run* $\mathsf{MRE.Setup}(N, \mathsf{pp} = (M, q)) \to (\mathsf{ek}, \{\mathsf{dk}_i^T\}) =: \mathsf{ak}$
- $\mathsf{Join}(\mathsf{ak}, \mathsf{id}) \to \mathsf{vk}_{\mathsf{id}}$*:* $\mathsf{vk}_{\mathsf{id}} := \mathsf{dk}_{\mathsf{id}}^T$
- $\mathsf{Auth}(\mathsf{ak}, \mathsf{m}, \mathcal{S}) \to \mathsf{cmd}_S$*: Randomly choose an* r*-dimensional column vector* $\boldsymbol{x}^T$ *from the domain of* f_{prm}*. Generate random small vectors* $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_N$ *from some distribution. Randomly choose a matrix* $CT \in \mathbb{Z}_q^{r \times M}$ *that satisfies* $CT \cdot \mathsf{dk}_i^T = \boldsymbol{x}^T + \boldsymbol{e}_i^T$ *for* $i \in \mathcal{S}$ *and* $CT \cdot \mathsf{dk}_i^T$ *is far from* $\boldsymbol{x}^T$ *for* $i \notin \mathcal{S}$*. Parse* CT *into the column vectors* $\mathsf{ct}_1^T, \ldots, \mathsf{ct}_M^T$*, encode them by* $F_i = f_{\mathsf{prm}}(\boldsymbol{ct}_i^T)$ *and the command is* $\mathsf{cmd}_S = (\mathsf{m} \otimes f_{\mathsf{prm}}(\boldsymbol{x}), F_1, \ldots, F_M)$*.*
- $\mathsf{Vrfy}(\mathsf{vk}_{\mathsf{id}}, \mathsf{cmd}_S) \to \mathsf{m}/\mathsf{reject}$*: For the device's key* $\mathsf{dk}_i := (d_{i,1}, \ldots, d_{i,M})$*, compute* $f = d_{i,1} \circ F_1 \otimes \cdots \otimes d_{i,M} \circ F_M$*, and* $\mathsf{m}' = \mathsf{m} \otimes f_{\mathsf{prm}}(\boldsymbol{x}) \otimes f^{-1}$*.*

For correctness, it is necessary to have some condition in f_{prm} and error vectors. For a legitimate command and decryption key

$$\begin{aligned} & d_{i,1} \circ F_1 \otimes \cdots \otimes d_{i,M} \circ F_M \\ & = f_{\mathsf{prm}}(d_{i,1}\boldsymbol{ct}_1^T + \cdots + d_{i,M}\boldsymbol{ct}_M^T) = f_{\mathsf{prm}}(CT \cdot \mathsf{dk}_i) = f_{\mathsf{prm}}(\boldsymbol{x}^T + \boldsymbol{e}_i^T) \end{aligned} \tag{2}$$

Thus, by the homomorphic property, $(\mathsf{m} \otimes f_{\mathsf{prm}}(\boldsymbol{x}^T)) \otimes f_{\mathsf{prm}}(\boldsymbol{x}^T + \boldsymbol{e}_i^T)^{-1} = \mathsf{m} \otimes f(\boldsymbol{e}_i^T)^{-1}$. A decoding mechanism is used to recover m from the above element.

In the discrete logarithm instantiation, we use $\boldsymbol{e}_i^T = 0$ for all i and the verification function returns m directly. On the other hand, in the lattice instantiation, $\boldsymbol{e}_i^T$ is a vector whose components are from $D_{\mathbb{Z},\sigma}$ with and a rounding function is used to recover.

For the device $i \notin \mathcal{S}$, the equation (2) computes

$$\mathsf{m} \otimes f_{\mathsf{prm}}(\boldsymbol{x} - \boldsymbol{x}') \tag{3}$$

for another random $\boldsymbol{x}'$ that is far from $\boldsymbol{x}^T$. Though it does not help to recover the message, the result (3) is possibly included in the domain of legitimate commands. To prevent the accident, it should use gimmicks separating the space of scalars into the legitimate commands and the others. We give an example of the separation in Section 6.2.

Below we discuss anonymity and unforgeability. We introduce a computational problem defined using f_{prm}, directly deduced from the anonymity game.

Its discrete logarithm version is reduced to the DDH problem. The lattice version is reduced to the decision LWE problem. Adding a strongly unforgeable signature, any ABA can be unforgeable.

Besides the anonymity and unforgeability, we also mention the security of the message and verification key. The message security is the hardness of recovering the message m from the command ciphertext without keys of target devices, but assume that the one has additional commands and keys from oracles. We have proved the message security in the discrete logarithm and the lattice instantiations are reduced to the hardness of message security of ElGamal encryption and the decision LWE problem, respectively. We postpone the proof to Appendix B because the original ABA definition does not consider it.

Also, key security is the hardness of recovering the key dk_i or its alternative $\mathsf{dk}_i'^T$ of the device i from other colluded keys $\{\mathsf{dk}_j^T\}$. It is easy to see that if one can recover an alternative key, it breaks anonymity. Thus, security proof of anonymity is also proof of key security.

4.2 Anonymity

We discuss the anonymity of the template construction. Note that it targets the version before the transformation. In the Step 4 of the anonymity game (Def. 4), the adversary can select $\mathsf{m}, \mathcal{S}, \mathcal{S}'$. Since we can cancel out m in the context of our template construction, breaking anonymity is the same as distinguishing the tuples

$$\begin{aligned}\mathsf{cmd}_S &= (f_{\mathsf{prm}}(\boldsymbol{x}^T), f_{\mathsf{prm}}(\boldsymbol{ct}_1^T), \ldots, f_{\mathsf{prm}}(\boldsymbol{ct}_M^T)) \text{ and}\\ \mathsf{cmd}_{S'} &= (f_{\mathsf{prm}}((\boldsymbol{x}')^T), f_{\mathsf{prm}}((\boldsymbol{ct}'_1)^T), \ldots, f_{\mathsf{prm}}((\boldsymbol{ct}'_M)^T)).\end{aligned}$$

We know there exists an index $\mathsf{id} \in \mathcal{S} \backslash \mathcal{S}'$ such that $\mathsf{dk}_{\mathsf{id}}$ can recover $f_{\mathsf{prm}}(\boldsymbol{x}^T + \boldsymbol{e}_i^T)$ via the relation (2).

We transform the above problem into a distinguishing problem between legitimate sequences and random sequences.

Definition 7. *($(f_{\mathsf{prm}}, \chi, M)$-linear distinguishing problem) For a function $f_{\mathsf{prm}}(\cdot)$ used in the template construction, consider the computational problem to distinguish the sequence*

$$(f_{\mathsf{prm}}(\boldsymbol{x}^T), f_{\mathsf{prm}}(\boldsymbol{c}_1^T), \ldots, f_{\mathsf{prm}}(\boldsymbol{c}_M^T)) \textit{ and } (f_{\mathsf{prm}}(\boldsymbol{r}^T), f_{\mathsf{prm}}(\boldsymbol{c}_1^T), \ldots, f_{\mathsf{prm}}(\boldsymbol{c}_M^T))$$

where $\boldsymbol{c}_1^T, \ldots, \boldsymbol{c}_M^T$ are randomly drawn from the domain of f_{prm}.

In the former case, $\boldsymbol{x}^T$ is computed $(\boldsymbol{c}_1^T || \boldsymbol{c}_2^T || \cdots || \boldsymbol{c}_M^T)\boldsymbol{d} + \boldsymbol{e}^T = \boldsymbol{c}_1^T d_1 + \cdots + \boldsymbol{c}_M^T d_M + \boldsymbol{e}^T$ by a fixed secret vector $\boldsymbol{d} = (d_1, \ldots, d_M)^T$ and a small random error $\boldsymbol{e}^T$ from χ^r. In the latter situation, $\boldsymbol{r}^T$ is random.

Theorem 1. *Using an adversary $\mathcal{A}$ that can win the anonymity game (Def. 4) with f_{prm} and noise distribution χ and dimension $2M$, it can solve the above distinguishing problem with parameters $(f_{\mathsf{prm}}, \chi, M)$ with high probability.*

Proof. Fix the parameters f_{prm}, χ and M, and suppose there exists an adversary $\mathcal{A}$. After the game setup of the anonymity game with $2M$ dimensions, the challenger generates a $(2M) \times (2M)$ random invertible matrix U.

In the collision query phase, suppose the adversary requires t verification keys; we can name them $\mathsf{dk}_1, \dots, \mathsf{dk}_t$ without loss of generality. Upon the queries, generate random M-dimensional vectors $\boldsymbol{r}_1, \dots, \boldsymbol{r}_t$ and set the fake verification keys to the adversary by $\mathsf{dk}_i^T = [(\boldsymbol{r}_i || \boldsymbol{u}_i)U]^T \in V^{2M}, i = 1, \dots, t$.

Also, using the virtual secret vector $\boldsymbol{d}$ of the linear distinguishing problem, define tentative decryption keys $\mathsf{dk}_i^T = [(\boldsymbol{d} || \boldsymbol{u}_i)U]^T$ for $i = t+1, \dots, M$, that are unknown by both the challenger and adversary. Upon requests from the adversary, the challenger sends the corrupted keys $\mathsf{dk}_1^T, \dots, \mathsf{dk}_t^T$.

In the authentication query phase, the challenger generates the command ciphertext of a query $(\mathsf{m}, \mathcal{S})$ as follows. Call the problem oracle and get an instance $(f_{\mathsf{prm}}(\boldsymbol{y}^T), f_{\mathsf{prm}}(\boldsymbol{c}_1^T), \dots, f_{\mathsf{prm}}(\boldsymbol{c}_M^T))$ where $\boldsymbol{y}^T$ is legitimate $\boldsymbol{x}^T$ or random $\boldsymbol{r}^T$. Denote $C = [\boldsymbol{c}_1^T || \cdots || \boldsymbol{c}_M^T]$. $F_i = f_{\mathsf{prm}}(\boldsymbol{c}_i)$ for $i = 1, \dots, M$. Then, for $i = M+1, \dots, M+t$, compute

$$f_{\mathsf{prm}}(C\boldsymbol{r}_i) = F_1 \circ r_{i,1} \otimes \cdots \otimes F_M \circ r_{i,M}$$

and

$$F_{M+i} := \begin{cases} f_{\mathsf{prm}}(C\boldsymbol{r}_i)^{-1} \otimes f_{\mathsf{prm}}(\boldsymbol{y}) \otimes f_{\mathsf{prm}}(\eta_i) & (i \in S) \\ f_{\mathsf{prm}}(rand) & (i \notin S) \end{cases}$$

where $rand$ means a random sampling from the domain of f_{prm}.

For $i = M+t+1, \dots, 2M$, compute

$$F_{M+i} := \begin{cases} f_{\mathsf{prm}}(\boldsymbol{y})^{-1} & (i \in S) \\ f_{\mathsf{prm}}(rand) & (i \notin S) \end{cases}$$

Then, compute

$$(V_1, \dots, V_{2M}) := (F_1, \dots, F_{2M})U^{-1}.$$

Here, the vector-matrix operations are performed with the operations $(\circ, \otimes)$, that is,

$$V_j = F_1 \circ u_{1,j} \otimes \cdots \otimes F_{2M} \circ u_{2M,j}$$

where $u_{i,j}$ is the (i,j)-element of U^{-1}. The command to the adversary is

$$\mathsf{cmd}_S = (\mathsf{m} \circ f_{\mathsf{prm}}(\boldsymbol{y}), V_1, \dots, V_{2M}).$$

It is easy to see that

$$\mathsf{Vrfy}(\mathsf{vk}_i, \mathsf{cmd}_S) = \mathsf{m} \otimes f_{\mathsf{prm}}(\boldsymbol{y}) \otimes f_{\mathsf{prm}}^{-1}(\boldsymbol{y}) = \mathsf{m} \otimes \begin{cases} f_{\mathsf{prm}}(\boldsymbol{\eta}_i) & i = 1, \dots, t \\ f_{\mathsf{prm}}(\boldsymbol{e}) & i = t+1, \dots, M \end{cases}$$

for $i \in \mathcal{S}$ if problem instance is legitimate.

However, if the problem instance is random, the relations on $i = t+1, \dots, M$ do not hold.

In the challenge query phase, for $(\mathsf{m}, \mathcal{S}_0, \mathcal{S}_1)$, the challenger returns $\mathsf{cmd}_{\mathcal{S}_b}$ for $b = 0$ or 1 in the same manner and checks the adversary's response. Checking the adversary's advantage, the challenger distinguishes the problem instance. □

Instantiations of Concrete Problems: We give instantiations of our problem (Definition 7) in the discrete logarithm and lattice situations.

In the discrete logarithm situation, we set $r = 1$ and $f_{\mathsf{prm}}(x) = g^x$ and noise variable is always zero. Thus, the problem is to distinguish

$$\{(g^x, g^{c_1}, \ldots, g^{c_M})\} \text{ and } \{(g^r, g^{c_1}, \ldots, g^{c_M})\}$$

where $(d_1, \ldots, d_M)$, $(c_1, \ldots, c_M)$ and r are a fixed secret vector, uniformly random vector, and a random number. In the legitimate situation, $x = c_1 d_1 + \cdots + c_M d_M$.

To our best knowledge, this problem is not well-known in the literature. However, this can be captured as a discrete logarithm version of the standard LWE problem and can be reduced to the DDH problem.

Proposition 1. *The above distinguishing problem can be reduced to DDH.*

Proof. First, we prove that an algorithm $\mathcal{A}_M$ to distinguish the above $M + 1$ dimensional vectors can solve the $M = 1$ problem. We can construct an algorithm $\mathcal{A}_1$ to solve the $M = 1$ problem. Before calling the oracle, $\mathcal{A}_1$ samples a virtual secrets $(d_2, \ldots, d_M)$ and fix them. Then, for a 1-instance (g^x, g^{c_1}), construct M-instance by $(g^y, g^{c_1}, \ldots, g^{c_M})$ with $g^y = g^x \cdot g^{c_2 d_2 + \cdots + c_M d_M}$ with randomly generated $c_2, \ldots, c_M$. If g^x is a legitimate $g^{c_1 d_1}$ (resp. random g^r), g^y is a legitimate (resp. random) number.

Next, we prove that $\mathcal{A}_1$ can distinguish DH instances. Fix a generator g and let (g^a, g^b, g^c) be a DDH sample where $c = ab$ or random. Note that for any randomly generated d_1, d_2, the relation $(g^{d_1} \cdot (g^b)^{d_2})^a = ((g^a)^{d_1} \cdot (g^c)^{d_2})$ holds if $c = ab$. Thus, a sequence of 1-instances with secret a can be generated by $(g^x, g^{c_1}) = (g^{d_1} \cdot (g^b)^{d_2}, (g^a)^{d_1} \cdot (g^c)^{d_2})$. If $c = ab$, it is a legitimate sample. On the other hand, if $c \neq ab$, the second component is $g^{c_1} = g^{ad_1 + abd_2} \cdot g^{(c-ab)d_2}$. Thus, the randomness of d_2 makes its distribution random. □

In the lattice situation, the problem is to distinguish

$$(\boldsymbol{p}\boldsymbol{x}^T, \boldsymbol{p}\boldsymbol{c}_1^T, \ldots, \boldsymbol{p}\boldsymbol{c}_M^T) \text{ and } (\boldsymbol{p}\boldsymbol{r}^T, \boldsymbol{p}\boldsymbol{c}_1^T, \ldots, \boldsymbol{p}\boldsymbol{c}_M^T)$$

where $\boldsymbol{x}^T$ is computed by $\sum_{i=1}^M d_i \boldsymbol{c}_i^T + \boldsymbol{e}^T$ by a secret vector $\boldsymbol{d} = (d_1, \ldots, d_M)$ and an error vector $\boldsymbol{e}^T$, and $\boldsymbol{r}^T$ is a random vector. This is the decision LWE problem.

4.3 Unforgeability

This section gives a transformation method to construct a t-unforgeable ABA. We first remark that the template construction does not have 1-unforgeability due to the homomorphic property. In fact, for

$$\mathsf{cmd}_{\mathcal{S}} = (\mathsf{m} \otimes f_{\mathsf{prm}}(\boldsymbol{x}^T), f_{\mathsf{prm}}(\mathsf{ct}_1^T), \ldots, f_{\mathsf{prm}}(\mathsf{ct}_M^T))$$

that targets $\mathcal{S}$ selected by an adversary having only $\mathsf{vk}_{\mathsf{id}}$, $\mathsf{cmd}_{\mathcal{S}} = (\mathsf{m} \otimes f_{\mathsf{prm}}(\boldsymbol{x}^T) \otimes \mathsf{c}, f_{\mathsf{prm}}(\mathsf{ct}_1^T), \ldots, f_{\mathsf{prm}}(\mathsf{ct}_M^T))$ is a legitimate ciphertext of the shifted message $\mathsf{m} \otimes \mathsf{c}$ accepted by some $\mathsf{id}' \in S \setminus \{\mathsf{id}\}$.

A simple transformation technique has been known from a CPA-secure public-key BE to a CCA1-secure one [8]. Following these notions and techniques, we construct our version of the transformation method from our template ABA to an unforgeable ABA.

Definition 8. *(Transformation)*
For an ABA scheme ABA = $(\mathsf{Setup}, \mathsf{Join}, \mathsf{Auth}, \mathsf{Vrfy})$ *and a strongly and existentially unforgeable signature* $\Sigma = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Vrfy})$*, the transformation of ABA, which we denote* ABA_{Σ} *is defined as follows.*

- $ABA_{\Sigma}.\mathsf{Setup}(1^{\lambda}, N, \mathcal{D}) \to (\mathsf{ak}, pk, sk)$*: Run* $ABA.\mathsf{Setup}(1^{\lambda}, N, \mathcal{D}) \to \mathsf{ak}$ *and* $\Sigma.\mathsf{KeyGen}(1^{\lambda}) \to (pk, sk)$*.*
- $ABA_{\Sigma}.\mathsf{Join}(\mathsf{ak}, \mathsf{id}) \to \mathsf{vk}_{\mathsf{id}}$*: Run the* $ABA.\mathsf{Join}$ *command and let* $\mathsf{vk}_{\mathsf{id}} = (ABA.\mathsf{vk}_{\mathsf{id}}, pk)$*.*
- $ABA_{\Sigma}.\mathsf{Auth}(\mathsf{ak}, \mathsf{m}, \mathcal{S}) \to (\mathsf{cmd}_{\mathcal{S}}, \sigma)$*: Execute* $ABA.\mathsf{Auth}(\mathsf{ak}, \mathsf{m}, \mathcal{S}) \to \mathsf{cmd}_{\mathcal{S}}$*. Generate the signature for the base command* $\Sigma.\mathsf{Sign}(sk, \mathsf{cmd}_{\mathcal{S}}) \to \sigma$*.*
- $ABA_{\Sigma}.\mathsf{Vrfy}(\mathsf{vk}_i, (\mathsf{cmd}_{\mathcal{S}}, \sigma)) \to \mathsf{m}/\mathsf{reject}$*: Check the signature* $\Sigma.\mathsf{Vrfy}(pk, \sigma, \mathsf{cmd}_{\mathcal{S}})$*. If the check fails, return* reject*. Passing the verifications, execute* $ABA.\mathsf{Vrfy}(\mathsf{vk}_{\mathsf{id}}, \mathsf{cmd}_{\mathcal{S}})$ *and return the result.*

Security proof is straightforward. In the security game (Definition 3), an adversary can get verification keys and $\{(\mathsf{cmd}_{\mathcal{S}}, \sigma)\}$ upon one's queries and suppose one can forge a command pair $(\mathsf{cmd}'_{\mathcal{S}'}, \sigma')$ with $(\mathsf{m}', \mathsf{id}')$ in Step 4. That is, $\Sigma.\mathsf{Vrfy}(\mathsf{pk}, \sigma', \mathsf{cmd}'_{\mathcal{S}'})$ returns accept and $ABA.\mathsf{Vrfy}(\mathsf{vk}_{\mathsf{id}'}, \mathsf{cmd}'_{\mathcal{S}'})$ returns m'.

The forging is splitting into two situations. If $\mathsf{cmd}'_{\mathcal{S}'}$ is not equal to any commands returned from the challenger in the authentication query step, $(\mathsf{cmd}'_{\mathcal{S}'}, \sigma')$ is a valid pair to break the strong unforgeability of the signature game, which is assumed to be hard.

On the other hand, consider the situation where $\mathsf{cmd}'_{\mathcal{S}'}$ is equal to $\mathsf{cmd}_{\mathcal{S}_a}$, one of returned commands in the authentication queries. We show this situation is impossible. Recall that the corresponding message m' and m_a in the commands cannot be equal by the requirement $\mathsf{m} \notin M_v$ in Step 3. Thus, the first element of $\mathsf{cmd}'_{\mathcal{S}'} = \mathsf{cmd}_{\mathcal{S}_a}$ is $\mathsf{m}' \otimes f_{\mathsf{prm}}(\boldsymbol{x}'^T) = \mathsf{m}_a \otimes f_{\mathsf{prm}}(\boldsymbol{x}_a^T)$ which are different representations of different messages. Thus, the verification results by $\mathsf{vk}_{\mathsf{id}'}$ must satisfy $ABA.\mathsf{Vrfy}(\mathsf{vk}_{\mathsf{id}'}, \mathsf{cmd}'_{\mathcal{S}'}) = \mathsf{m}'$ and $ABA.\mathsf{Vrfy}(\mathsf{vk}_{\mathsf{id}'}, \mathsf{cmd}_{\mathcal{S}_a}) = \mathsf{m}_a$. This contradicts to the requirement $\mathsf{m}' \neq \mathsf{m}_a$ and $\mathsf{cmd}'_{\mathcal{S}'} = \mathsf{cmd}_{\mathcal{S}_a}$.

Therefore, forging a command ciphertext is hard due to the strong unforgeability of the signature.

5 Concatenation of ABAs

The sequential concatenation of small-size ABAs is a simple way to reduce the length of ciphertexts by restricting choice of target devices.

Definition 9. *(Sequential concatenation of ABAs) Consider K ABAs. Let them be $ABA_j = (\mathsf{Setup}_j, \mathsf{Join}_j, \mathsf{Auth}_j, \mathsf{Vrfy}_j)$ and N_j be the maximum number of devices controlled by the j-th ABA. The concatenated ABA is defined as follows. Each device is indicated by a vector* $\mathsf{id} = (i_1, \ldots, i_K)$.

- $ABA.\mathsf{Setup}$*: Execute* $ABA_j.\mathsf{Setup}(1^\lambda, N, \mathcal{D}) \to \mathsf{ak}_j$ *for all* $j \in [K]$*. The order of execution does not matter and let* $\mathsf{ak} := \{\mathsf{ak}_1, \ldots, \mathsf{ak}_K\}$.
- $ABA.\mathsf{Join}(\mathsf{ak}, \mathsf{id})$*: For* $\mathsf{id} = (i_1, \ldots, i_K)$*, execute* $ABA_j.\mathsf{Join}(\mathsf{ak}_j, i_j) \to \mathsf{vk}_{j,i_j}$*. The verification key is the concatenation* $\mathsf{vk}_{\mathsf{id}} = (\mathsf{vk}_{1,i_1}, \ldots, \mathsf{vk}_{K,i_K})$.
- $ABA.\mathsf{Auth}(\mathsf{ak}, \mathsf{m}, \mathcal{S})$*: The set of target devices is indicated by* $\mathcal{S} = \mathcal{S}_1 \times \cdots \times \mathcal{S}_K$ *where* $\mathcal{S}_j \subset [N_j]$*. The command ciphertext* $\mathsf{cmd}_{\mathcal{S}}$ *is also the concatenation of* $\mathsf{cmd}_{\mathcal{S}_j} = \mathsf{Auth}(\mathsf{ak}_j, \mathsf{m}, \mathcal{S}_j)$ *for* $j = 1, \ldots, K$*, and broadcast it.*
- $ABA.\mathsf{Vrfy}(\mathsf{vk}_{\mathsf{id}}, \mathsf{cmd}_{\mathcal{S}}) \to \mathsf{m}/\mathsf{reject}$*. Check whether* $\mathsf{Vrfy}(\mathsf{vk}_{j,i_j}, \mathsf{cmd}_{\mathcal{S}_j})$ *for all* j*. If all the verification has been accepted, output* m*, if otherwise, output* reject.

Since we have proved the base ABA has anonymity from the computational problems, the concatenated ABA also has anonymity on two sets of limited forms. That is, $\mathcal{S}_1 \times \mathcal{S}_2 \times \cdots \times \mathcal{S}_K$ and $\mathcal{S}_1' \times \mathcal{S}_2 \times \cdots \times \mathcal{S}_K$ are indistinguishable if all $\mathcal{S}_2, \ldots, \mathcal{S}_K$ are equivalent. However, one can forge a ciphertext and break anonymity by rearranging components when some devices collude via the following example. To prevent such rearranging attacks, we modify the concatenated scheme in the following subsection.

Example 1. Consider the concatenation of $K = 2$ ABAs with $N_1 = N_2 = 2$. The composed ABA can control $N_1 N_2 = 4$ devices and we name them by $\mathsf{id} = (1,1), (1,2), (2,1)$ and $(2,2)$.

Suppose that $\mathsf{id} = (1,1)$ and $\mathsf{id} = (2,2)$ are colluded, that is, an attacker have $\mathsf{vk}_{1,1} = (\mathsf{vk}_{1,1}, \mathsf{vk}_{2,1})$ and $\mathsf{vk}_{2,2} = (\mathsf{vk}_{1,2}, \mathsf{vk}_{2,2})$. The one can generate other verification keys $\mathsf{vk}_{12} = (\mathsf{vk}_{1,1}, \mathsf{vk}_{2,2})$ and $\mathsf{vk}_{21} = (\mathsf{vk}_{1,2}, \mathsf{vk}_{2,1})$ via the recombination of components. Thus, the attacker can recover any legitimate ciphertext and know the target devices.

For instance, in the anonymity game, let the challenger's set be $\mathcal{S}_0 = \{(1,2)\}$ and $\mathcal{S}_1 = \{(2,1)\}$ which satisfies the condition (1). Then $\mathsf{cmd}_{\mathcal{S}_b}$ can be easily verified to distinguish.

Another situation where one can forge a ciphertext is possible. Suppose a device is honest-but-curious, and it receives ciphertexts $\mathsf{cmd}_{1,1} = (\mathsf{cmd}_{S_1}, \mathsf{cmd}_{S_2})$ and $\mathsf{cmd}_{2,2} = (\mathsf{cmd}_{S_1'}, \mathsf{cmd}_{S_2'})$ whose targets are $\{(1,1)\}$ and $\{(2,2)\}$, respectively. It can construct a forged ciphertext $(\mathsf{cmd}_{S_1}, \mathsf{cmd}_{S_2'})$ targeting $\mathsf{id} = (1,2)$ if the attacker knows the ciphertexts contain the same message.

5.1 Modification Against Recombination Attack

The simple concatenated ABA does not have anonymity and unforgeability by rearranging colluded keys. Also, a non-target device can recover the message. To prevent recombination attacks, we employ two methods. The first idea is to distribute m into K shares and recover it in a target device via the homomorphic property of f_{prm}. The other idea is from Agrawal et al.'s inner product encryption

[13]. In the context of this paper, we employ the mechanism to increase the amount of information to break the anonymity security by adding variables.

Definition 10. *(A template construction against simple recombination attacks)*

- $\mathsf{Setup}(1^\lambda, K, \mathcal{D}) \to \mathsf{ak}$*: Fix a prime field* F_q *and a dimension* M *of base MREs. Execute* $\mathsf{MRE}_j.\mathsf{KeyGen}(\mathsf{pp}) \to (\mathsf{ek}_j, \{\mathsf{dk}^T_{j,i}\}_{i\in[N_j]})$ *for* $j \in [K]$*, where* $\mathsf{ek}_j := \{\mathsf{dk}^T_{j,i}\}$*. Generate random matrices* $A_{j,i} \overset{\$}{\leftarrow} \mathbb{Z}_q^{r\times M}$ *for* $j \in [K], i \in [N_j]$*, a random invertible matrix* $W \in \mathbb{Z}_q^{2M\times 2M}$ *and a vector* $\boldsymbol{u} \in \mathbb{Z}_q^r$*. The key is* $\mathsf{ak} = (\{\mathsf{dk}^T_{j,i}\}, \{\mathsf{ek}_j\}, \{A_{j,i}\}, W, \boldsymbol{u})$*.*
- $\mathsf{Join}(\mathsf{ak}, \mathsf{id}) \to \mathsf{vk}_{\mathsf{id}}$*: For a device* $\mathsf{id} = (i_1, i_2, \dots, i_K)$*, generate a random vector* $\mathsf{uk}_{\mathsf{id}}$ *such that* $\sum_{j=1}^K A_{j,i_j}\mathsf{uk}^T_{\mathsf{id}} = \boldsymbol{u} \pmod q$*. Then, define the verification key by*

$$\mathsf{vk}_{\mathsf{id}} = (W(\mathsf{dk}_{1,i_1}||\mathsf{uk}_{\mathsf{id}})^T, \dots, W(\mathsf{dk}_{K,i_K}||\mathsf{uk}_{\mathsf{id}})^T).$$

- $\mathsf{Auth}(\mathsf{ak}, \mathsf{m}, \mathcal{S}) \to \mathsf{cmd}_{\mathcal{S}}$*: Suppose the target devices are indicated by* $\mathcal{S}_1 \times \dots \times \mathcal{S}_K \subset \prod[N_j]$*. Pick random vectors* $\boldsymbol{t}_j^T \in \mathbb{Z}_q^r$ *and let* $\boldsymbol{x}^T := \boldsymbol{t}_1^T + \dots + \boldsymbol{t}_K^T$*. Generate random matrices* $CT_{j,i}$ $(j \in [M], i \in [N_j])$ *such that*

$$CT_{j,i} \cdot \mathsf{dk}_{j,\ell} = \begin{cases} \boldsymbol{t}_j^T + \boldsymbol{e}_{j,i}^T & (i = \ell \text{ and } i \in S_j) \\ rand & (i \notin S_j) \end{cases}$$

where rand represents a random far from $\boldsymbol{t}_j^T$*. Define the matrix* $C_{j,i} := (CT_{j,i}||A_{j,i})W^{-1}$ *and split it into the* $2M$ *column vectors by* $C_{j,i} = (\boldsymbol{c}^T_{j,i,1}||\cdots||\boldsymbol{c}^T_{j,i,2M})$*. Then, the command ciphertext* $\mathsf{cmd}_{\mathcal{S}}$ *is* $\mathsf{m} \otimes f_{\mathsf{prm}}(\boldsymbol{x}^T + \boldsymbol{u}^T)$ *and the sequence* $\{f_{\mathsf{prm}}(\boldsymbol{c}^T_{j,i,\ell})\}$*.*
- $\mathsf{Vrfy}(\mathsf{vk}_{\mathsf{id}}, \mathsf{cmd}_{\mathcal{S}})$*: For* $\mathsf{id} = (i_1, \dots, i_K)$*, denote* $\mathsf{vk}_{\mathsf{id}} = (\boldsymbol{v}_1, \dots, \boldsymbol{v}_K)$ *and let the* ℓ*-th element of* $\boldsymbol{v}_j$ *be* $v_{j,\ell}$*. For the command* $(\mathsf{m}\otimes f_{\mathsf{prm}}(\boldsymbol{x}^T+\boldsymbol{u}^T), \{F_{j,i,\ell}\})$*, compute the sum* $T_j = \sum_{\ell=1}^{2M} v_{j,\ell} \otimes F_{j,i_j,\ell}$ *and* $\mathsf{m} \otimes f_{\mathsf{prm}}(\boldsymbol{x}^T + \boldsymbol{u}^T) \otimes (T_1 \otimes \dots \otimes T_K)^{-1}$*.*

The correctness is immediate as follows. For a target $\mathsf{id} = (i_1, \dots, i_K)$,

$$\begin{aligned} T_j &= \sum_{\ell=1}^{2M} v_{j,\ell} \otimes f_{\mathsf{prm}}(\boldsymbol{c}^T_{j,i_j,\ell}) = f_{\mathsf{prm}}\left(\sum_{ell=1}^{eM} \boldsymbol{c}^T_{j,i_j,\ell} v_{j,\ell}\right) \\ &= f_{\mathsf{prm}}((CT_{j,i_j}||A_{j,i_j})W^{-1} \cdot W(\mathsf{dk}_{j,i_j}||\mathsf{uk}_{\mathsf{id}})) = f_{\mathsf{prm}}(\boldsymbol{t}_j^T + \boldsymbol{e}^T_{j,i} + A_{j,i_j}\mathsf{uk}_{\mathsf{id}}) \end{aligned}$$

and the sum in the sense of $\otimes$ is

$$T_1 \otimes \dots \otimes T_K = f_{\mathsf{prm}}\left(\sum_{j=1}^K \boldsymbol{t}_j^T + \boldsymbol{u}^T + \sum_{j=1}^K \boldsymbol{e}_{j,i_j}\right).$$

Therefore, it recovers $\mathsf{m} \otimes f_{\mathsf{prm}}\left(\sum_{j=1}^K \boldsymbol{e}^T_{j,i_j}\right)$.

The unforgeability is realized by adding a signature as in Section 4.3. We discuss the anonymity.

5.2 On the Anonymity

We emphasize that we can guarantee the above modification has t-anonymity with a limited range of t, though a more complicated heuristic obfuscation would increase t by sacrificing resources. As we will discuss in Appendix A, under the situation where the freedom in the choice of $\mathcal{S}$ is limited, a pattern of $\mathsf{id} \overset{?}{\in} \mathcal{S}$ obtained by an attacker can reveal information on whether $\mathsf{id}' \overset{?}{\in} \mathcal{S}$ for other id's even if the cryptographic gadgets work completely and the standard anonymity (Definition 4) is satisfied. Therefore, we think constructing a more complicated scheme to achieve stronger anonymity is not a good strategy for practical purpose. As the next best thing, we discuss the security via the relation between the number of variables and equations in algebraic systems.

To distinguish two commands $\mathsf{cmd}_{\mathcal{S}}$ and $\mathsf{cmd}_{\mathcal{S}'}$ that contain the same message, it is necessary to find whether the sum $\sum_{j=1}^{K}\sum_{\ell=1}^{2M} v_{j,\ell} \otimes f_{\mathsf{prm}}(\boldsymbol{c}_{j,i_j,\ell}^T)$ nears to $f_{\mathsf{prm}}(\boldsymbol{x}^T + \boldsymbol{u}^T)$ for an unknown verification key $\mathsf{vk}_{\mathsf{id}}$. Here, we discuss a necessary number of colluded keys and authentication queries to reveal $\mathsf{vk}_{\mathsf{id}}$.

Suppose an attacker wants to distinguish two commands from two sets $\mathcal{S}_0$ and $\mathcal{S}_1$ so that $\mathsf{id} \in \mathcal{S}_0 \triangle \mathcal{S}_1$ for id selected by the attacker. Also, assume that the attacker's distinguishing is via recovering the verification key $\mathsf{vk}_{\mathsf{id}}$. Let $(1, 1, \ldots, 1)$ be the id without loss of generality. Split the matrix W into the upper and lower matrices with $M \times 2M$ dimensions:

$$W = \begin{bmatrix} W_1 \\ W_2 \end{bmatrix}$$

Denote $\boldsymbol{w}_{j,i} = W_1 \mathsf{dk}_{j,i}^T$ and $\boldsymbol{u}_{\mathsf{id}} = W_2 \mathsf{uk}_{\mathsf{id}}^T$. Then, the verification key that one wants to recover is written as $\mathsf{vk}_{\mathsf{id}} = (\boldsymbol{w}_{1,1} + \boldsymbol{u}_{\mathsf{id}}, \ldots, \boldsymbol{w}_{K,1} + \boldsymbol{u}_{\mathsf{id}})$. Here, $\boldsymbol{u}_{\mathsf{id}}$ also satisfies $\sum_{j=1}^{K} A_{j,1} \mathsf{uk}_{\mathsf{id}}^T = \boldsymbol{u}^T$ for unknown matrices $A_{j,1}$.

Thus, the number of variables to fix is KM for $\boldsymbol{w}_{j,1}$ $(j = 1, \ldots, K)$, $K \cdot rM$ for $A_{j,i}$, $2M^2$ for W_2 and r for $\boldsymbol{u}$. Also, the number of unknown variables in $\mathsf{uk}_{\mathsf{id}}$ is r since the other $M - r$ variables can be random by construction. With a new colluded key $\mathsf{vk}_{\mathsf{id}'}$, MK equations can be obtained and it introduces new variables on $\boldsymbol{w}_{j,i}$ and $A_{j,i}$. For the authentication query, it does not introduce new equations due to the random variables $\boldsymbol{t}_j$ in construction.

To minimize the number of unknown variables, we minimize the range of indexes. For $\mathsf{id} = (i_1, \ldots, i_N)$ that satisfies $i_j \in [2]$ for $j = 1, \ldots, s$ and $i_j = 1$ for $i = s+1, \ldots, K$, $t = 2^s - 1$ colluded keys are possible. Here, the number of variables are $2(K+s)M$ for $\boldsymbol{w}_{j,i_j}$, $2(K+s) \cdot rM$ for A_{j,i_j}, $2M^2$ for W_2, tr for $\mathsf{uk}_{\mathsf{id}}$, r for $\boldsymbol{u}$. Therefore the total number of variables is totally, $V = 2M^2 + 2(K+s)rM + 2(K+s)M + tr + r$.

After getting t colluded keys, the problem is to solve simultaneous equations with $2M^2 + 2(K+s)rM + 2(K+s)M + tr + r$ variables and tKM equations. It is necessary to satisfy at least

$$\begin{aligned} & 2M^2 + 2(K+s)rM + 2(K+s)M + tr + r < tKM \\ \Leftrightarrow\ & t > \frac{2M^2 + 2(K+s)rM + 2(K+s)M + r}{KM - r}. \end{aligned}$$

to fix a unique solution. The left-hand side is bounded by $M/K+2r+2$. Therefore, it has evidence of anonymity against $2+2r$ corruption.

For the situation $r=1$, the lower bound is only four corruption. However, the situation we use in this paper is the discrete logarithm construction, which might be secure by the one-wayness of the discrete logarithm. On the other hand, in the lattice situation, r is the number of samples in LWE, and it should be greater than 900, which guarantees the security against collusion of 1800 devices. We think they are practically secure in the both situations.

6 Concrete Schemes and Security Parameters

This section gives concrete schemes based on discrete logarithms and lattices, security parameters, and rough estimations of communication costs. Both construction assumes $N_j = 2$ for all j.

We should discuss a relation between the space of an ABA command and cryptographic message. Let us denote $\mathcal{C}$ space of ABA commands and denote $\mathcal{M}$ be of the messages. A target device can recover m, whereas a nontarget device gets a random number which is possibly interpreted as a legitimate command. It can cause a serious error. To prevent accidents, we should employ a gimmick in the authentication algorithm. Here, note that a simple signature to the command may be useless because a nontarget device can know the command by the exhaustive search when the command space is small.

In the discrete logarithm construction, the probability of the accident, that is, the situation where a random number in $\mathbb{Z}_q$ is in the message space, is exponentially small since we should take q exceeds 2^{2000}. On the other hand, in the lattice construction, we should sophisticate it more carefully.

6.1 Discrete Logarithm Construction

We instantiate the discrete logarithm construction by setting $f_{\mathsf{prm}}(a) = g^a \bmod q$ to Definition 8 and 10.

Definition 11. *(Discrete Logarithm Construction)*
• $\mathsf{Setup}(1^\lambda, K, \mathcal{D}) \to \mathsf{ak}$*: Fix a prime field* $\mathbb{Z}_q$ *and a dimension* M *of base MREs. Execute* $\mathsf{MRE}_j.\mathsf{KeyGen}(\mathsf{pp}) \to (\mathsf{ek}_j, \{\mathsf{dk}_{j,i}^T\}_{i\in[N_j]})$ *for* $j \in [K]$ *where* $\mathsf{ek}_j := \{\mathsf{dk}_{j,i}^T\}$*. Generate random vectors* $\boldsymbol{a}_{j,i} \xleftarrow{\$} \mathbb{Z}_q^M$ *for* $j \in [K], i = 1, 2$*, a random invertible matrix* $W \in \mathbb{Z}_q^{2M\times 2M}$ *and a constant* $u \in \mathbb{Z}_q$*. Execute* $\Sigma.\mathsf{KeyGen}(1^\lambda) \to (pk, sk)$ *The key is* $\mathsf{ak} = (\{\mathsf{dk}_{j,i}^T\}, \{\mathsf{ek}_j\}, \{\boldsymbol{a}_{j,i}\}, W, u, pk, sk)$*.*
• $\mathsf{Join}(\mathsf{ak}, \mathsf{id}) \to \mathsf{vk}_{\mathsf{id}}$*: For a device* $\mathsf{id} = (i_1, i_2, \ldots, i_K)$*, generate a random vector* $\mathsf{uk}_{\mathsf{id}}$ *such that* $\sum_{j=1}^K \boldsymbol{a}_{j,i_j} \mathsf{uk}_{\mathsf{id}}^T = u \pmod q$*. Then, put the verification key by*

$$\mathsf{vk}_{\mathsf{id}} = \{(W(\mathsf{dk}_{1,i_1}||\mathsf{uk}_{\mathsf{id}})^T, \ldots, W(\mathsf{dk}_{K,i_K}||\mathsf{uk}_{\mathsf{id}})^T), pk\}.$$

• $\mathsf{Auth}(\mathsf{ak}, \mathsf{m}, \mathcal{S}) \to \mathsf{cmd}_{\mathcal{S}}$*: Suppose the target devices are indicated by* $\mathcal{S}_1 \times \cdots \times \mathcal{S}_K \subset [2]^K$*. Pick random numbers* $t_j \in F_q$ *and let* $x := t_1 + \cdots + t_K$*. Generate* *random vectors* $\boldsymbol{ct}_{j,i}$ $(j \in [M], i \in [2])$ *such that*

$$\boldsymbol{ct}_{j,i} \cdot \mathsf{dk}_{j,\ell}^T = \begin{cases} t_j & (i = \ell \textit{ and } i \in S_j) \\ rand & (i \notin S_j) \end{cases}$$

where rand represents an output from random number generator except for t_j*.*

Define the vector $\boldsymbol{c}_{j,i} := (\boldsymbol{ct}_{j,i}^T || \boldsymbol{a}_{j,i})W^{-1}$ *and parse it into the coordinates by* $\boldsymbol{c}_{j,i} = (c_{j,i,1}, \ldots, c_{j,i,2M})$*. Then, the command* $\mathsf{cmd}_{\mathcal{S}}$ *is* $m \cdot g^{x+u}$ *and the sequence* $\{g^{c_{j,i,\ell}}\}$*. Finally, generate a signature to the command* $\Sigma.\sigma(sk, \mathsf{cmd}_{\mathcal{S}}) \to \sigma$*. Ciphertext is the pair* $(\mathsf{cmd}_{\mathcal{S}}, \sigma)$*.*

- $\mathsf{Vrfy}(\mathsf{vk}_{\mathsf{id}}, (\mathsf{cmd}_{\mathcal{S}}, \sigma))$*: Check the signature by* $\Sigma.\mathsf{Vrfy}(pk, \sigma, \mathsf{cmd}_{\mathcal{S}})$ *at first and if it returns* reject*, it stops with returning* reject*. Otherwise, continue the process. For* $\mathsf{id} = (i_1, \ldots, i_K)$*, denote the vector part of* $\mathsf{vk}_{\mathsf{id}}$ *be* $(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_K)$ *and let the* ℓ*-th element of* $\boldsymbol{v}_j$ *be* $v_{j,\ell}$*. For the command* $(m \cdot g^{x+u}, \{g^{c_{j,i,\ell}}\})$*, compute* $T_j = \prod_{\ell=1}^{2M} (g^{c_{j,i_j,\ell}})^{v_{j,\ell}}$ *and* $m \cdot g^{x+u} \cdot (T_1 \cdot \cdots \cdot T_K)^{-1}$*.*

The correctness and security are given in the template construction. We explain the communication cost. Remember that K, M, q are the parameter of the number of concatenated ABAs, vector dimensions, and the modulus. We also denote pksize and sigsize the sizes of public key and signature for unforgeability. The size of $\mathsf{vk}_{\mathsf{id}}$ is $2MK \log_2 q + \mathsf{pksize}$ [bits] since it consists K vectors of $2M$ dimension in $\mathbb{Z}_q$ and the public key of the signature. Each command consists $1 + K \cdot 2 \cdot 2M = 1 + 4KM$ numbers in $\mathbb{Z}_q$ and thus, the communication cost is $(1 + 4KM) \log_2 q + \mathsf{sigsize}$.

Since we take $N_j = 2$ for all j, it is necessary to take $M \geq 2$ for decryption keys. $M \geq 3$ is suitable to introduce some randomness in the keys. We use $M = 3$. Since it can control 2^K devices, about $K = 20 - 30$ is enough to control $10^6 - 10^9$ devices in practice. For the security of DLP, $\log_2 q \geq 2048$ is required [14, Sect. 5.5.1.1]. Also, for the security of ECDLP, we assume Curve25519 spends 256 bits to store a point on the curve in the compressed representation. Table 1 gave the summary.

6.2 LWE-based Construction

We instantiate the LWE-based construction by setting $f_{\boldsymbol{p}}(\boldsymbol{x}) = \boldsymbol{x}\boldsymbol{p}^T \bmod q$ to Definition 8 and 10. M, K, r are parameters. Also, we use Q to the multiple of plaintext to avoid the effect of noises. Concretely, plaintext $\overline{\mathsf{m}}$ is an integer such that $0 \leq \overline{\mathsf{m}} < q/Q$ and embed it in the form of $\mathsf{m} = \overline{\mathsf{m}} \cdot Q$ in the command ciphertext. In verification, the device rounds the decoded message m' to $\overline{\mathsf{m}'} = \lfloor \mathsf{m}'/Q \rceil$. Also, we use an integer L to distinguish a legitimate and a nonlegitimate command. If the verification function returns $\overline{\mathsf{m}} < L$, it is interpreted as a legitimate command and executes it. If otherwise, return the reject symbol. In our construction, we assume $q > 2KLQ$ to separate the legitimate commands and the rejecting messages.

Definition 12. *(LWE-based Construction)*

• $\mathsf{Setup}(1^\lambda, K, \mathcal{D}) \to \mathsf{ak}$*: Fix a prime field* F_q *and a dimension* M *of base MREs. Execute* $\mathsf{MRE}_j.\mathsf{KeyGen}(\mathsf{pp}) \to (\mathsf{ek}_j, \{\mathsf{dk}_{j,i}^T\}_{i\in[2]})$ *for* $j \in [K]$*, where* $\mathsf{ek}_j := \{\mathsf{dk}_{j,i}^T\}$*. Generate random matrices* $A_{j,i} \xleftarrow{\$} \mathbb{Z}_q^{r\times M}$ *for* $j \in [K], i \in [2]$*, a random invertible matrix* $W \in \mathbb{Z}_q^{2M\times 2M}$ *and a vector* $\boldsymbol{u} \in \mathbb{Z}_q^r$*. Execute* $\Sigma.\mathsf{KeyGen}(1^\lambda) \to (pk, sk)$ *The key is* $\mathsf{ak} = (\{\mathsf{dk}_{j,i}\}, \{\mathsf{ek}_j\}, \{\boldsymbol{a}_{j,i}\}, W, u, pk, sk)$*.*
• $\mathsf{Join}(\mathsf{ak}, \mathsf{id}) \to \mathsf{vk}_{\mathsf{id}}$*: For a device* $\mathsf{id} = (i_1, i_2, \ldots, i_K)$*, generate a random vector* $\mathsf{uk}_{\mathsf{id}}$ *such that* $\sum_{j=1}^K A_{j,i_j}\mathsf{uk}_{\mathsf{id}}^T = \boldsymbol{u} \pmod q$*. Then, define the verification key by*

$$\mathsf{vk}_{\mathsf{id}} = \{(W(\mathsf{dk}_{1,i_1}||\mathsf{uk}_{\mathsf{id}})^T, \ldots, W(\mathsf{dk}_{K,i_K}||\mathsf{uk}_{\mathsf{id}})^T), pk\}.$$

• $\mathsf{Auth}(\mathsf{ak}, \mathsf{m} \in \mathcal{M}, \mathcal{S}) \to \mathsf{cmd}_{\mathcal{S}}$*: Suppose the target devices are indicated by* $\mathcal{S}_1 \times \cdots \times \mathcal{S}_K \subset \prod[2]$*. Pick random vectors* $\boldsymbol{t}_j^T$ *so that* $\boldsymbol{pt}_j^T \in \{LQ, \ldots, 2LQ-1\}$ *and let* $\boldsymbol{x}^T := \boldsymbol{t}_1^T + \cdots + \boldsymbol{t}_K^T$*. Here,* $\boldsymbol{px}^T$ *is greater than* KLQ *since there is no overflow in* $\mathbb{Z}_q$ *by the condition* $q > 2KLQ$*.*

Generate random matrices $CT_{j,i}$ $(j \in [M], i \in N_j)$ *such that*

$$CT_{j,i} \cdot \mathsf{dk}_{j,\ell}^T = \begin{cases} \boldsymbol{t}_j^T + \boldsymbol{e}_{j,i}^T & (i = \ell \text{ and } i \in S_j) \\ \boldsymbol{z}_{j,i} & (i \notin S_j) \end{cases} \tag{4}$$

where $\boldsymbol{z}_{j,i}$ *is a random vector such that* $\boldsymbol{pz}_{j,i}^T$ *is less than* LQ*.*

Define the matrix $C_{j,i} := (CT_{j,i}||A_{j,i})W^{-1}$ *and split it into the* $2M$ *column vectors by* $C_{j,i} = (\boldsymbol{c}_{j,i,1}^T||\cdots||\boldsymbol{c}_{j,i,2M}^T)$*. Let the command part be* $(\mathsf{m}\cdot Q + \boldsymbol{p}(\boldsymbol{x}^T + \boldsymbol{u}^T), \{\boldsymbol{pc}_{j,i,\ell}^T\})$ *and generate its signature* σ*. Then, the command* $\mathsf{cmd}_{\mathcal{S}}$ *is the pair of the above command and* σ*.*
• $\mathsf{Vrfy}(\mathsf{vk}_{\mathsf{id}}, \mathsf{cmd}_{\mathcal{S}})$*: Check the signature by* $\Sigma.\mathsf{Vrfy}(pk, \sigma, \mathsf{cmd}_{\mathcal{S}})$ *at first and if it returns* reject*, it stops with returning* reject*. If the signature is valid, execute the decryption process as follows. For* $\mathsf{id} = (i_1, \ldots, i_K)$*, denote the vector part of* $\mathsf{vk}_{\mathsf{id}}$ *be* $(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_K)$ *and let the* ℓ*-th element of* $\boldsymbol{v}_j$ *be* $v_{j,\ell}$*. For a command* $(\mathsf{m} + \boldsymbol{p}(\boldsymbol{x}^T + \boldsymbol{u}^T), \{\boldsymbol{pc}_{j,i,\ell}^T\})$*, compute the sum* $T_j = \sum_{\ell=1}^{2M} v_{j,\ell}F_{j,i_j,\ell}$ *and* $\overline{\mathsf{m}'} = \mathsf{m} + \boldsymbol{p}(\boldsymbol{x}^T + \boldsymbol{u}^T) - (T_1 + \cdots + T_K)$*. Decode the message by* $\mathsf{m}' = \lfloor \overline{\mathsf{m}'}/Q \rceil$*. If it is greater than* L*, return* reject *and if otherwise, return* m'*.*

The correctness and securities are already given in the template construction.

We should give detail on distinguishing the legitimate command. In the computation of $\overline{\mathsf{m}'}$, we have

$$\boldsymbol{p}(\boldsymbol{x}^T + \boldsymbol{u}^T) - (T_1 + \cdots T_K) = \boldsymbol{px}^T - \sum_{j=1}^K \boldsymbol{p}(CT_{j,i_j} \cdot \mathsf{dk}_{j,i_j}). \tag{5}$$

after cancelling $\boldsymbol{u}^T$. Here, each factor is $\boldsymbol{p}(\boldsymbol{t}_j^T + \boldsymbol{e}_{j,i}^T)$ or $\boldsymbol{pz}_{j,i}$ by (4). By the conditions $\boldsymbol{px}^T \geq KLQ$ and $\boldsymbol{pz}_{j,i} < LQ$, if there is a factor from $\boldsymbol{pz}_{j,i}$, the sum is greater than LQ and the resulting $\overline{\mathsf{m}'}$ is greater than L.

We describe the processing resources. As in the discrete situation, we assume $N_j = 2$ for all j and K is about $20 - 30$. From Theorem 1 (with instantiation to decision LWE), the hardness of the decision LWE problem with parameters

(n, m, q, σ) is the security base of the anonymity of lattice-based ABA with parameters $M = 2n$, modulus q, and the error parameter σ. Thus, we take the parameters $M = 2n$ and the same (q, σ) so that the LWE problem is intractable.

On the other hand, it should reduce the probability of decoding errors by modifying q and Q. Concretely, following the detail of recovering $\overline{\mathsf{m}'}$ in the verification, the noises are

$$\boldsymbol{p}(\boldsymbol{x}^T + \boldsymbol{u}^T) - (T_1 + \cdots T_K) = \sum_{j=1}^{K} \boldsymbol{p} \cdot \boldsymbol{e}_{j,i_j}^T. \tag{6}$$

Assuming each coordinates of $\boldsymbol{e}_{j,i_j}^T$ are continuous Gaussian $N(0, \sigma^2)$, the distribution of the error is $N(0, K||\boldsymbol{p}||^2\sigma^2)$. For the extremely low probability case, the following bound gives a good approximation

$$\Pr[|N(0, s^2)| \geq \beta] = 1 - \operatorname{erf}(\beta/s) < \exp\left(-\beta^2/s^2\right).$$

Thus, taking β so that the bound is very small, it derives a bound of the error in practice. For example, take the error bound by the inverse of $10^9 \cdot 2^{32} \cdot 2^{64} \approx 8\dot{}10^{37} \approx e^{87.3}$, whose factors are the number of controlled devices, the number of seconds in 100 years, and safety margins. This derives $\beta > \sqrt{87.3}s$. Therefore, we can assume the absolute value of (6) is smaller than $\sqrt{87.3} \cdot \sqrt{2K}||\boldsymbol{p}||\sigma$ in practice. Since $\boldsymbol{p}$ works as a secret vector of LWE in the security proof, it should be a discrete Gaussian [17] and its derivation is σ. As the same argument, we can assume $||\boldsymbol{p}|| < \sqrt{87.3}\sqrt{M}\sigma = 2\sqrt{2 \cdot 87.3n}\sigma$ and thus we take Q so that $\sqrt{87.3} \cdot \sqrt{2K}\sigma \cdot 2\sqrt{2 \cdot 87.3n}\sigma \approx 350\sqrt{Kn}\sigma^2 < Q$.

As an example situation, we set $K = 20$ for controlling a million devices, and set $\sigma = 3$. Let the space of legitimate message space be 4 (two bits). Then, q is a prime larger than $2KLQ > 2LK \cdot 350\sqrt{Kn}\sigma^2 \approx 2253956\sqrt{n}$. To achieve 128-bit security in ABA. We use Albrecht et al.'s lattice estimator [16] as of May 2022, and obtain the dimension 926.

For another set, we summarize the parameter in Table 2 and Table 3. Since one verification key and a command ciphertext consists of $2KM = 4Kn$ and $1 + 4KM = 1 + 8Kn$ elements of $\mathbb{Z}_q$ respectively, the sizes in bytes are the smallest integers greater than $(4Kn)\lfloor \log_2 q \rfloor/8$ and $(1 + 8Kn)\lfloor \log_2 q \rfloor/8$.

7 Concluding remarks

We proposed a template construction of ABA that can control an exponential number of devices compared to the length of ciphertexts. Our design rationale is taking an edge of conditions from the trilemma among the security, ciphertext length, and freedom in the choice of target devices. Then we achieved ABA with logarithmic-order ciphertexts by restricting the third condition, namely the choice of controllable devices.

Acknowledgement This research was in part conducted under a contract of "Research and development on IoT malware removal / make it non-functional technologies for effective use of the radio spectrum" among "Research and Development for Expansion of Radio Wave Resources (JPJ000254)", which was supported by the Ministry of Internal Affairs and Communications, Japan. This work was in part supported by JSPS KAKENHI Grant Number JP22H03590.

References

[1] Yohei Watanabe, Naoto Yanai, and Junji Shikata. "Anonymous Broadcast Authentication for Securely Remote-Controlling IoT Devices". In: *Advanced Information Networking and Applications*. Springer International Publishing, 2021, pp. 679–690.

[2] Aggelos Kiayias and Katerina Samari. "Lower Bounds for Private Broadcast Encryption". In: *Information Hiding*. Ed. by Matthias Kirchner and Dipak Ghosal. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 176–190. ISBN: 978-3-642-36373-3.

[3] Yohei Watanabe Hirokazu Kobayashi and Junji Shikata. "Asymptotically Tight Lower Bounds in Anonymous Broadcast Encryption and Authentication". In: *Cryptography and Coding - 18th IMA International Conference, IMACC 2021, Proceedings*.

[4] Attrapadung Nuttapong. "Unified Frameworks for Practical Broadcast Encryption and Public Key Encryption with High Functionalities". PhD thesis. 2007.

[5] Adam Barth, Dan Boneh, and Brent Waters. "Privacy in Encrypted Content Distribution Using Private Broadcast Encryption". In: *Financial Cryptography and Data Security*. Ed. by Giovanni Di Crescenzo and Avi Rubin. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 52–64. ISBN: 978-3-540-46256-9.

[6] Benoît Libert, Kenneth G. Paterson, and Elizabeth A. Quaglia. "Anonymous Broadcast Encryption: Adaptive Security and Efficient Constructions in the Standard Model". In: *Public Key Cryptography – PKC 2012*. Ed. by Marc Fischlin, Johannes Buchmann, and Mark Manulis. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 206–224. ISBN: 978-3-642-30057-8.

[7] Nelly Fazio and Irippuge Milinda Perera. "Outsider-Anonymous Broadcast Encryption with Sublinear Ciphertexts". In: *Public Key Cryptography - PKC 2012 - 15th International Conference on Practice and Theory in Public Key Cryptography, Darmstadt, Germany, May 21-23, 2012. Proceedings*. Ed. by Marc Fischlin, Johannes Buchmann, and Mark Manulis. Vol. 7293. Lecture Notes in Computer Science. Springer, 2012, pp. 225–242. DOI: 10.1007/978-3-642-30057-8_14. URL: https://doi.org/10.1007/978-3-642-30057-8_14.

[8] Jiwon Lee et al. "Combinatorial Subset Difference – IoT-Friendly Subset Representation and Broadcast Encryption". In: *Sensors* 20.11 (2020). ISSN: 1424-8220. DOI: 10.3390/s20113140. URL: https://www.mdpi.com/1424-8220/20/11/3140.

[9] *FrodoKEM Learning With Errors Key Encapsulation Algorithm Specifications And Supporting Documentation (June 4, 2021).* Website: `https://frodokem.org/files/FrodoKEM-specification-20210604.pdf`.

[10] *Falcon: Fast-Fourier Lattice-based Compact Signatures over NTRU Specification v1.2 – 01/10/2020.* Website: `https://falcon-sign.info/falcon.pdf`.

[11] Kaoru Kurosawa et al. "Some Bounds and a Construction for Secure Broadcast Encryption". In: *Advances in Cryptology - ASIACRYPT '98.* Ed. by Kazuo Ohta and Dingyi Pei. Vol. 1514. Lecture Notes in Computer Science. Springer, 1998, pp. 420–433.

[12] Mihir Bellare et al. "Multirecipient Encryption Schemes: How to Save on Bandwidth and Computation Without Sacrificing Security". In: *IEEE Transactions on Information Theory* 53.11 (2007), pp. 3927–3943. DOI: `10.1109/TIT.2007.907471`.

[13] Shweta Agrawal, David Mandell Freeman, and Vinod Vaikuntanathan. "Functional Encryption for Inner Product Predicates from Learning with Errors". In: *Advances in Cryptology – ASIACRYPT 2011.* Ed. by Dong Hoon Lee and Xiaoyun Wang. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 21–40. ISBN: 978-3-642-25385-0.

[14] Elaine Barker et al. *Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography.* Website: `https://nvlpubs.nist.gov/nistpubs/SpecialPublications/nist.sp.800-56Ar3.pdf`.

[15] Oded Regev. "On Lattices, Learning with Errors, Random Linear Codes, and Cryptography". In: *Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing.* STOC '05. Baltimore, MD, USA: Association for Computing Machinery, 2005, pp. 84–93. ISBN: 1581139608. DOI: `10.1145/1060590.1060603`. URL: `https://doi.org/10.1145/1060590.1060603`.

[16] *Estimate all the LWE, NTRU schemes!* Website: `https://estimate-all-the-lwe-ntru-schemes.github.io/docs/`.

[17] Richard Lindner and Chris Peikert. "Better Key Sizes (and Attacks) for LWE-Based Encryption". In: *CT-RSA*. Ed. by Aggelos Kiayias. Vol. 6558. Lecture Notes in Computer Science. Springer, 2011, pp. 319–339. ISBN: 978-3-642-19073-5.

A Anonymity from a Single Command

In the notion of standard anonymity (Definition 4), the situation where no information leakage except for the corrupted devices is defined via the indistinguishability on two sets $\mathcal{S}_0$ and $\mathcal{S}_1$. We think the validity of this definition bases on the assumption that the target set is uniformly chosen from $2^{[N]}$ and information leakage always results from cryptographic vulnerability.

On the other hand, we emphasize that there is a possibility that information is leaked regardless of the cryptographic vulnerability if freedom in the target devices selection is limited. We explain a situation below.

Example 2. Consider an ABA whose id's are indicated by two dimensional vectors (i_1, i_2) and $i_j = 1, 2$. Thus, total of four devices are controlled. Suppose the target ids are indicated by $\mathcal{S} = \mathcal{S}_1 \times \mathcal{S}_2, \mathcal{S}_j \subset [2]$, which is equivalent to the family of set of ids indicated by $d_1 d_2 \in \{0, 1, *\}^2$ where $*$ is the wild-card. Totally, nine patterns are possible as in Table 4.

Table 4. Selection of target devices.

(d_1, d_2)	(1,1)	(1,2)	(2,1)	(2,2)
(1,1)	●			
(1,2)		●		
(1,*)	●	●		
(2,1)			●	
(2,2)				●
(2,*)			●	●
(*,1)	●		●	
(*,2)		●		●
(*,*)	●	●	●	●

Consider the situation where $\mathsf{id} = (1, 2)$ and $(2, 1)$ are colluded and one can know only $\mathsf{vk}_{1,2}$ and $\mathsf{vk}_{2,1}$ by a protection against recombination attack. Suppose the attacker knows a command ciphertext targets both $\mathsf{id} = (1, 2)$ and $(2, 1)$. Then, from the Table 4, $d_1 d_2 = (*, *)$ is revealed and it can know $\mathsf{id} = (1, 1)$ and $(2, 2)$ are also target devices without any attacks on cryptographic components.

To formulate this kind of anonymity, we introduce the notion of single anonymity. This insecurity is independent of the security of cryptographic components.

Definition 13. *(Single Anonymity) Suppose $\mathcal{D}$ be a family of sets $\mathcal{S} \subset 2^{[N]}$ that can be specified as a target of ABA. We say it has a single anonymity if a pattern of $\{\mathsf{Vrfy}(\mathsf{vk}_{\mathsf{id}}, \mathsf{cmd}_{\mathcal{S}})\}_{\mathsf{id} \in W}$ does not reveal whether $\mathsf{id}' \in \mathsf{cmd}_{\mathcal{S}}$ for $\mathsf{id}' \notin W$.*

Proposition 2. *If the freedom in the target devices selection, it does not have the single anonymity.*

Proof. Suppose the selection of target devices has restricted freedom. That is, assume there is a subset $\mathcal{S} \subset [N]$ that cannot be specified as a target. We separate the situation. First, in the situation where there exists a $\mathcal{S}' \supset \mathcal{S}$ that can be specified as a target. Take the smallest $\mathcal{S}$ that satisfies the condition. Then, suppose all the devices $\mathcal{S} \cup ([N] \setminus \mathcal{S}')$ has been colluded and receive a command $\mathsf{cmd}_{\mathcal{S}'}$ that targets $\mathcal{S}'$. The adversary can verify the commands in the infected devices and can know for $\mathsf{id} \in \mathcal{S}$ (resp. $\mathsf{id} \in [N] \setminus \mathcal{S}'$), they are in target (resp. non-target.) With the restriction of the choice of $\mathcal{S}$, the adversary also obtain that the devices in $\mathcal{S}' \setminus \mathcal{S}$ are in the target without using any cryptographic attacks.

Next, consider the situation where for any subset $\mathcal{S} \subset [N]$ that cannot be specified as a target, a subset $\mathcal{S}' \supset \mathcal{S}$ does not exist. Let $\mathcal{S}$ be a non-target set and take $\bar{\mathcal{S}}$ as a maximal target set so that $\bar{\mathcal{S}} \subset \mathcal{S}$. We can show the information leakage with a similar argument. Suppose all the devices $\bar{\mathcal{S}} \cup ([N] \setminus \mathcal{S})$ has been colluded and receive a command $\mathsf{cmd}_{\mathcal{S}}$ that targets $\mathcal{S}$. The adversary can verify the commands in the infected devices and can know for $\mathsf{id} \in \bar{\mathcal{S}}$ (resp. $\mathsf{id} \in [N] \backslash \mathcal{S}$), they are in target (resp. non-target.) With the restriction of the choice of $\mathcal{S}$, the adversary also obtain that the devices in $\mathcal{S}' \setminus \mathcal{S}$ are not in the target. □

We think one of our future works is to investigate the relationship between the restriction of target devices choice (that bounds ciphertext length) and the strength of anonymity.

B Message Security

This section discusses the hardness of distinguishing messages from the command ciphertexts without keys of target devices.

We introduce a game-based definition of message indistinguishability.

Definition 14. *(IND-CPA security of ABA with t-collusion) Consider the following game between an adversary* A *and a challenger* C.

0: Share N the number of participant devices and pp *the public parameter.* C *runs* $\mathsf{Setup}(1^\lambda, N, \mathcal{D}) \to \mathsf{ak} = (\mathsf{ek}, \{\mathsf{dk}_i\})$.

1: (Collusion query) A *selects* $\mathsf{id} \in [N]$ *and send it to the challenger.* C *runs* $\mathsf{Join}(\mathsf{ak}, \mathsf{id}) \to \mathsf{vk}_{\mathsf{id}}$. *Add* id *to W and send* $\mathsf{vk}_{\mathsf{id}}$ *to* A. A *can repeat this step until the number of colluded devices is less than t.*

2: (Challenge ciphertext) A *selects a set of target devices* $\mathcal{S} \subset [N], \mathcal{S} \cap W = \phi$ *and two messages* $\mathsf{m}_0, \mathsf{m}_1$ *and send them to* C. C *generates a ciphertext* $\mathsf{cmd}_{\mathcal{S}}$ *from* m_b *where b is randomly chosen from* $\{0, 1\}$ *and return the ciphertext to* A.

3: (Encryption query) A *can ask* C *to encrypt* $(\mathsf{m}', \mathcal{S}')$ *and receive the ciphertext* $\mathsf{cmd}_{\mathcal{S}'}$

4: A *guesses b' from* $\mathsf{cmd}_{\mathcal{S}}$.

The advantage of A *is defined by* $2 \cdot |\Pr[b = b'] - 1/2|$. *The scheme is said to be IND-CPA secure if the advantage is negligible.*

In our template construction, a challenge ciphertext has a form

$$\mathsf{cmd}_{\mathcal{S}} = (\mathsf{m}_b \otimes f_{\mathsf{prm}}(\boldsymbol{x}^{\boldsymbol{T}}), f_{\mathsf{prm}}(\boldsymbol{c}_1^T), \ldots, f_{\mathsf{prm}}(\boldsymbol{c}_M^T))$$

and also a command ciphertext with known m' in the encryption query phase is

$$\mathsf{cmd}_{\mathcal{S}'} = (\mathsf{m}' \otimes f_{\mathsf{prm}}(\boldsymbol{x}^T), f_{\mathsf{prm}}((\boldsymbol{c}_1')^T), \ldots, f_{\mathsf{prm}}((\boldsymbol{c}_M')^T)).$$

Theorem 2. *The IND-CPA security of ABA in the discrete logarithm instantiation can be reduced to the DDH assumption.*

Proof. Suppose $\mathcal{A}$ can distinguish the command ciphertext of ABA with discrete logarithm instantiation, working with M dimensional vectors, N devices, $f_g(a) = g^a$ and t colluded devices. We show how the challenger distinguishes an instance of DDH assumption (g, g^r, g^x, g^c) where $c = rx$ or a random number.

Let $\overline{\mathsf{dk}_i}^T, i \in [N]$ be fake decryption keys of base MRE that the challenger randomly generates. Upon the adversary's collusion queries, it returns a sequence of the fake keys. Upon the adversary's set $\mathcal{S}$, generate a random number $\overline{x}$ and a row vector CT so that

$$CT \cdot \overline{\mathsf{dk}_i}^T = \begin{cases} \quad \overline{x} \quad (i \in \mathcal{S}) \\ rand \; (i \notin \mathcal{S}) \end{cases}$$

where $rand$ is a random number except for $\overline{x}$. Then, parse $CT = (\mathsf{ct}_1, \ldots, \mathsf{ct}_M)$.

Upon the messages $\mathsf{m}_0, \mathsf{m}_1$, choose m_b randomly and let the returning command be

$$\mathsf{cmd}_{\mathcal{S}'} = (\mathsf{m}_b \cdot g^c, g^{r\mathsf{ct}_1}, \ldots, g^{r\mathsf{ct}_M}).$$

Let dk_i^Ts be the true keys assumed to be used in a virtual encryption system from the adversary's view. It satisfies $CT \cdot \mathsf{dk}_i^T = x$. The relation between the fake and true keys are $\overline{\mathsf{dk}_i}^T = \mathsf{dk}_i^T \cdot (\overline{x}/x)$.

Then, upon the adversary's encryption query $(\mathsf{m}', \mathcal{S}')$, again generate a random number $\overline{x'}$ and a matrix CT' so that

$$CT' \cdot \overline{\mathsf{dk}_i}^T = \begin{cases} \quad \overline{x'} \quad (i \in \mathcal{S}) \\ rand \; (i \notin \mathcal{S}) \end{cases}$$

and the command ciphertext be

$$\mathsf{cmd}_{\mathcal{S}'} = (\mathsf{m}' \cdot g^{x \cdot (\overline{x'}/\overline{x})}, g^{\mathsf{ct}'_1}, \ldots, g^{\mathsf{ct}'_M}).$$

Since we have g^x, $\overline{x}$ and $\overline{x'}$ the first element is easily computable. For the true key $\mathsf{dk}_i^T = (d_{i,1}, \ldots, d_{i,M})^T$, we have

$$\prod_{i=1}^{M} (g^{\mathsf{ct}'_i})^{d_{i,j}} = g^{CT' \cdot \mathsf{dk}_i} = g^{\overline{\mathsf{dk}_i}^T \cdot (x/\overline{x})} = g^{\overline{x'} \cdot (x/\overline{x})}$$

Thus, this is a legitimate command that encrypts m'.

Finally, the adversary guesses b' and the advantage $|\Pr[b = b'] - 1/2|$ should be high if $c = rx$ and small if c is random. □

Besides the discrete logarithm situation, we connect the security of ABA to the hardness of the decision LWE in the lattice case.

Theorem 3. *Let (n, q, m) be the LWE parameter, and suppose the noise distribution is from χ. The hardness of the decision LWE guarantees the security of ABA against CPA with lattice instantiation with noise distribution $\chi + \chi$ and vector dimension $M = 2n$. Also, assume the domain of $f_{\mathsf{prm}}(\cdot)$ has dimension r.*

Proof. Let $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_r$ and $\boldsymbol{b}^T = (b_1, \ldots, b_r)^T$ be sample instances of the decision LWE. Here, $b_i = \boldsymbol{a}_i \boldsymbol{s}^T + e_i \bmod q$ or a random in $\mathbb{Z}_q$. We also let $\boldsymbol{e}^T = (e_1, \ldots, e_r)$ from the noise distribution χ^r.

Let $\mathcal{A}$ be an adversary that can distinguish two messages of ABA working with $2n$ dimensional vectors, N devices, and t colluded devices. The template function $f_{\boldsymbol{p}}(\boldsymbol{x}^T) = \boldsymbol{p}\boldsymbol{x}^T$ is defined over r dimensional vectors and $\boldsymbol{p}$ is assumed to be a small vector to be kept secret but random for security.

For collusion queries, we can assume the ID's $\mathsf{id} = 1, \ldots, t$ w.l.o.g, and also we let fake verification keys by $2n$ dimensional vector $\overline{\mathsf{vk}}_i^T = [(\boldsymbol{r}_i||\boldsymbol{u}_i)U]^T$ where $\boldsymbol{r}_i$s and U are random vectors from $\mathbb{Z}_q^n$ and a random matrix in $\mathbb{Z}_q^{2n\times 2n}$, respectively. Also, for other IDs, we tentatively put $\overline{\mathsf{vk}}_i^T = [(\boldsymbol{s}||\boldsymbol{u}_i)U]^T$ which is unknown for both challenger and adversary since $\boldsymbol{s}$ is the secret vector of the LWE instance.

Upon the adversary's request $\mathsf{m}_0, \mathsf{m}_1$ and $\mathcal{S}$ in Step 2, the challenge ciphertext is constructed by using LWE samples $\boldsymbol{a}_i, \boldsymbol{b}$ and randomly generated m_b as follows. For the row vectors $\boldsymbol{a}_i$, construct the $r \times n$ matrix and parse it into n column vectors

$$\begin{bmatrix} \boldsymbol{a}_1 \\ \vdots \\ \boldsymbol{a}_r \end{bmatrix} = \left[\, A_1^T | \cdots | A_n^T \,\right]$$

and define A_i the transpose of A_i^T. Then, the command is

$$\mathsf{cmd}_{\mathcal{S}} = (m_b + \boldsymbol{b}\boldsymbol{p}^T, [A_1\boldsymbol{p}^T, \ldots, A_n\boldsymbol{p}^T, F_{n+1}, \ldots, F_{2n}]U^{-1} \bmod q).$$

Here, the latter elements are defined by

$$F_{n+i} = \begin{cases} \quad rand & (i \notin \mathcal{S}) \\ \quad 0 & (i \in \mathcal{S} \cap \{t+1, \ldots, N\}) \text{ //non-colluded keys} \\ \boldsymbol{b}\boldsymbol{p}^T - \sum_{i=1}^n r_i A_i \boldsymbol{p}^T + \boldsymbol{\eta_i}\boldsymbol{p}^T & (i \in \mathcal{S} \cap [t]) \text{ //colluded keys} \end{cases}$$

where $\boldsymbol{\eta}_i$ is a vector from $D_{\mathbb{Z},\sigma}^r$. For this setting, the results of the target and nontarget device's verifications are as follows. For $i \in S \cap [t]$, we have

$$\begin{aligned} &(A_1\boldsymbol{p}^T, \ldots, A_n\boldsymbol{p}^T, F_{n+1}, \ldots, F_{2n})U^{-1} \cdot (\overline{\mathsf{vk}_i}^T) \\ &= (A_1\boldsymbol{p}^T, \ldots, A_n\boldsymbol{p}^T, F_{n+1}, \ldots, F_{2n}) \cdot (\boldsymbol{r}_i||\boldsymbol{u}_i)^T = \boldsymbol{b}\boldsymbol{p}^T + \boldsymbol{\eta}_i\boldsymbol{p}^T. \end{aligned}$$

Thus,

$$\mathsf{Vrfy}(\overline{\mathsf{vk}}_i, \mathsf{cmd}_{\mathcal{S}}) = m_b - \boldsymbol{\eta}_i\boldsymbol{p}^T \tag{7}$$

For $i \in S \cap ([N] \setminus [t])$, we have

$$\begin{aligned} &(A_1\boldsymbol{p}^T, \ldots, A_n\boldsymbol{p}^T, F_{n+1}, \ldots, F_{2n})U^{-1} \cdot (\overline{\mathsf{vk}_i}^T) \\ &= (A_1\boldsymbol{p}^T, \ldots, A_n\boldsymbol{p}^T, F_{n+1}, \ldots, F_{2n}) \cdot (\boldsymbol{s}_i||\boldsymbol{u}_i)^T = \sum_{i=1}^n s_i \boldsymbol{a}_i \boldsymbol{p}^T \end{aligned}$$

Thus,

$$\mathsf{Vrfy}(\overline{\mathsf{vk}}_i, \mathsf{cmd}_{\mathcal{S}}) = \mathsf{m}_b + \boldsymbol{b}\boldsymbol{p}^T - \sum_{i=1}^n s_i \boldsymbol{a}_i \boldsymbol{p}^T = \mathsf{m} + (\boldsymbol{e} + \boldsymbol{\eta}_i)\boldsymbol{p}^T. \tag{8}$$

For the encryption queries $(\mathsf{m}', \mathcal{S}')$, it can construct the command in the same manner.

Therefore, LWE samples can be converted to the instance of ABA. If the LWE instance is legitimate (resp. random), the advantage of $\mathcal{A}$ should be high (resp. low). Thus, $\mathcal{A}$ can solve the decision LWE problem.

Finally, it is necessary to keep attention to the noise distribution. In the above transformation, the noises (7) and (8) are slightly different. Replacing both $\boldsymbol{e}^T + \boldsymbol{\eta}_i^T$ and $\boldsymbol{\eta}_i^T$ by $(\boldsymbol{\eta}')_i^T$ which has larger derivations, the hardness of distinguishing is amplifying. If one takes $\boldsymbol{e}^T, \boldsymbol{\eta}_i^T$ are discrete Gaussians of variance σ^2, $(\boldsymbol{\eta}')_i^T$ should be the discrete Gaussians of variance $2\sigma^2$. We completee the proof of the relation between noise parameters. □