




# Fine-grained Secure Attribute-based Encryption

Yuyu Wang<sup>1</sup> , Jiaxin Pan<sup>2</sup> , and Yu Chen<sup>3,4,5</sup> 

<sup>1</sup> University of Electronic Science and Technology of China, Chengdu, China  
[wangyuyu@uestc.edu.cn](mailto:wangyuyu@uestc.edu.cn)

<sup>2</sup> Department of Mathematical Sciences,  
NTNU - Norwegian University of Science and Technology, Trondheim, Norway  
[jiaxin.pan@ntnu.no](mailto:jiaxin.pan@ntnu.no)

<sup>3</sup> School of Cyber Science and Technology,  
Shandong University, Qingdao 266237, China

<sup>4</sup> State Key Laboratory of Cryptology, P.O. Box 5159, Beijing 100878, China

<sup>5</sup> Key Laboratory of Cryptologic Technology and Information Security, Ministry of  
Education, Shandong University, Qingdao 266237, China  
[yuchen@sdu.edu.cn](mailto:yuchen@sdu.edu.cn)

**Abstract.** Fine-grained cryptography is constructing cryptosystems in a setting where an adversary’s resource is a-prior bounded and an honest party has less resource than an adversary. Currently, only simple form of encryption schemes, such as secret-key and public-key encryption, are constructed in this setting.

In this paper, we enrich the available tools in fine-grained cryptography by proposing the *first* fine-grained secure attribute-based encryption (ABE) scheme. Our construction is adaptively secure under the widely accepted worst-case assumption,  $\text{NC}^1 \not\subseteq \oplus\text{L}/\text{poly}$ , and it is presented in a generic manner using the notion of predicate encodings (Wee, TCC’14). By properly instantiating the underlying encoding, we can obtain different types of ABE schemes, including identity-based encryption. Previously, all of these schemes were unknown in fine-grained cryptography. Our main technical contribution is constructing ABE schemes without using pairing or the Diffie-Hellman assumption. Hence, our results show that, even if one-way functions do not exist, we still have ABE schemes with meaningful security. For more application of our techniques, we construct an efficient (quasi-adaptive) non-interactive zero-knowledge (QA-NIZK) proof system.

**Keywords.** Fine-grained cryptography, identity-based encryption, attribute-based encryption, quasi-adaptive non-interactive zero-knowledge proof.

## 1 Introduction

### 1.1 Motivation

Modern cryptography bases the security of schemes on assumptions, including the basic ones (such as the existence of one-way functions (OWFs)), the more

advanced ones (such as the hardness of factoring, discrete logarithms, and some lattice problems), and the much more exotic ones (such as the existence of generic groups [29,25] or algebraic groups [17]). Although there is some analysis on these assumptions, it is less desirable. We are interested in how to construct cryptography based on much mild assumptions or which form of security cryptography can be achieved if all classical assumptions (such as the existence of OWFs) do not hold.

Fine-grained cryptography is a direction in approaching the aforementioned problems. It aims at cryptography with weaker security in a setting where adversaries have only bounded resources and honest users have less resources than the adversaries. Under this setting it is possible to make the underlying assumption extremely mild, for instance, assuming  $\text{NC}^1 \subsetneq \oplus\text{L}/\text{poly}$ . This is a widely accepted worst-case assumption. As  $\oplus\text{L}/\text{poly}$  is the class of languages with polynomial-sized branching programs and all languages in  $\text{NC}^1$  have polynomial-sized branching programs of constant width [3], this assumption holds if there exists one language having only polynomial-sized branching programs of non-constant width. This is different to assuming the existence of OWFs which is an average-case assumption. It requires that the OWF be hard to invert on a random input. Hence,  $\text{NC}^1 \subsetneq \oplus\text{L}/\text{poly}$  is more likely to be true.

The study on fine-grained cryptography was initialized by Merkle [26]. In the recent years, we are interested in which kind of cryptosystems can be constructed in this setting. We highlight the recent constructions of OWFs [8], symmetric-key and (additively homomorphic) public-key encryption [13,9], hash proof systems (HPS) [14], and non-interactive zero-knowledge (NIZK) proof systems [2]. However, due to the restriction on running resources, many important primitives remain unknown. Surprisingly, digital signature schemes are among them, although they are implied by OWFs in the classical setting.

**Our goal: fine-grained secure ABEs.** We focus on constructing attribute-based encryption (ABE) schemes [19] with fine-grained security, since it has many applications and implies important primitives, including digital signatures. In an ABE scheme, messages are encrypted under descriptive values  $x$ , secret keys are associated with values  $y$ , and a secret key decrypts the ciphertext if and only if  $p(x, y) = 1$  for some Boolean predicate  $p$ . Here the predicate  $p$  may express arbitrary access policy. This is in contrast to traditional public-key encryption (PKE) schemes without access control on data. Identity-based encryption [28,6,12] is a simplified version of ABE, where  $p$  is the equality predicate, and it implies signatures in a natural manner (even in the fine-grained setting).

In general, it is challenging to construct ABEs. For instance, in the classical setting, it is shown that IBEs cannot be constructed using trapdoor permutations (TDP) or CCA-secure PKE schemes in a black-box manner [7]. Moreover, many pairing-based constructions of ABE and IBE (for instance, [10,5]) heavily rely on the algebraic structures of pairing groups. These necessary structures are not available in fine-grained cryptography. Thus, in this paper, we will transform the state of the art of fine-grained cryptography, which only provides primitives related to TDP and CCA-secure PKE, and develop new tools to achieve our goal.

## 1.2 Our Contributions

We construct the *first* fine-grained secure ABE scheme. In particular, our scheme is computable in  $AC^0[2]$  and secure against adversaries in  $NC^1$ . Note that  $AC^0[2] \subsetneq NC^1$  [27,30]. Similar to several existing  $NC^1$  fine-grained primitives [13,9,14], the security of our scheme is based on the same worst-case assumption  $NC^1 \subsetneq \oplus L/poly$ . This is a widely accepted, weak assumption. For simplicity, we consider fine-grained cryptography as schemes with  $NC^1$  honest users and adversaries and security based on  $NC^1 \subsetneq \oplus L/poly$  in the rest of this paper.

Previously, fine-grained cryptography can only achieve symmetric-key and public-key encryption and HPS. Our work enriches its available tools and brings fine-grained cryptography closer to classical cryptography in terms of functionality.

In particular, our construction is presented in a generic manner using predicate encodings [32,10]. Hence, by suitably instantiating the underlying encoding, we directly obtain a fine-grained IBE scheme (which in turn implies a fine-grained signature scheme), fine-grained ABEs for inner-product encryption, non-zero inner-product encryption, spatial encryption, doubly spatial encryption, boolean span programs, and arithmetic span programs, and also fine-grained broadcast encryption and fuzzy IBE schemes. Prior to this work, it was unknown whether these primitives can be constructed in  $NC^1$  based on a worst-case complexity assumption.

Finally, we use our technique to construct an efficient quasi-adaptive NIZK [23] with fine-grained security. Here “quasi-adaptive” means that common reference strings may depend on the language of the NIZK system.

## 1.3 Technique Overview

We borrow the frameworks of the pairing-based constructions of IBEs in [5] and ABEs in [10] to upgrade the available fine-grained techniques [22,1,14] in achieving our goal. At a high-level point of view, the main idea in [5,10] is to find a suitable symmetric-key primitive and transform it to the corresponding public-key scheme using pairings and the Matrix Decisional Diffie-Hellman (MDDH) assumption [16]. More precisely, the Blazy-Kiltz-Pan (BKP) framework [5] transforms message authentication codes (MAC) to IBEs, and the Chen-Gay-Wee (CGW) framework [10] transforms predicate encodings to ABEs.

However, the goal of fine-grained cryptography is to construct schemes with mild assumptions other than the MDDH assumption. Our work develops techniques to build ABEs without pairings or the MDDH assumption, but only under the mild assumption that  $NC^1 \subsetneq \oplus L/poly$ . For simplicity, we mostly focus on our techniques in the context of IBE here, and give some ideas about how they can be extended to construct ABEs. In this paper, we consider adaptive security where adversaries can adaptively request user secret keys and a challenge ciphertext.

**The approach of BKP and its limitations in  $NC^1$ .** The “MAC $\rightarrow$ IBE” transformation of BKP [5] is an abstraction of the Chen-Wee (CW) IBE scheme [11],

and it also generalizes the “PRF→Signature” framework by Bellare and Goldwasser (BG) [4] in the IBE context. The BKP transformation requires an “affine MAC”, namely, a MAC whose verification is done by checking a particular system of affine equations. Variables in these affine equations are included in the MAC secret key, and the (public) coefficients are derived from the message (which will be the identity of the resulting IBE scheme) to be signed. Such a MAC scheme can be constructed based on the Diffie-Hellman assumption which is generalized as the MDDH assumption.

We give some ideas about how an affine MAC can be turned into an IBE scheme. The master public key of an IBE scheme,  $\mathbf{pk} = \text{Com}(\text{sk}_{\text{MAC}})$ , is a commitment of the MAC secret key,  $\text{sk}_{\text{MAC}}$ . A user secret key  $\text{usk}[\text{id}]$  of an identity  $\text{id}$  consists of a BG signature, namely, a MAC tag  $\tau_{\text{id}}$  on the message  $\text{id}$  and a NIZK proof of the validity of  $\tau_{\text{id}}$  w.r.t. the secret key committed in  $\mathbf{pk}$ .

Since the MAC verification consists of only affine equations, after implementing the aforementioned commitments and NIZK proofs with the (tuned) Groth-Sahai (GS) proof system [20], the BKP IBE ciphertext  $\text{ct}_{\text{id}}$  can be viewed as a randomized linear combination of  $\mathbf{pk}$  w.r.t.  $\text{id}$ . This is the key observation of BKP. The BKP framework can be further improved and extended to construct ABEs using predicate encodings [32] as in the CGW framework [10].

The MDDH assumption and the pairing-based GS proofs are two key ingredients for the BKP framework which are not available in fine-grained cryptography. One direction to resolve this is to develop a fine-grained GS proof system, but it is not clear what the counterpart of “pairing-product equations” will be. Instead, we achieve our goal with a simpler and more direct approach.

**A hard subset membership problem for  $\text{NC}^1$  circuits.** We first need to find a counterpart of the MDDH assumption in  $\text{NC}^1$ , since the separation assumption  $\text{NC}^1 \not\subseteq \oplus\text{L}/\text{poly}$  does not directly give us tools in constructing cryptographic schemes. In the work of [22,1], it is shown that, if  $\text{NC}^1 \not\subseteq \oplus\text{L}/\text{poly}$  holds, then the following two distributions are identical for  $\text{NC}^1$  circuits:

$$\underbrace{\{\mathbf{M}_0 \in \{0, 1\}^{n \times n} : \mathbf{M}_0 \stackrel{s}{\leftarrow} \text{ZeroSamp}(n)\}}_{=D_0} \quad \text{and} \quad \underbrace{\{\mathbf{M}_1 \in \{0, 1\}^{n \times n} : \mathbf{M}_1 \stackrel{s}{\leftarrow} \text{OneSamp}(n)\}}_{=D_1}$$

where  $n = n(\lambda)$  is some polynomial in security parameter  $\lambda$ , and the randomized sampling algorithms `ZeroSamp` and `OneSamp` output matrices with rank  $n - 1$  and full rank, respectively. Concrete definitions of these algorithms are given in Section 2.2, and they are not relevant in this section.

This indistinguishability implies a hard subset membership problem in  $\text{NC}^1$  implicitly given by Egashira, Wang, and Tanaka [15] for their HPS: Given a matrix  $\mathbf{M}$  from  $D_0$  and a random vector  $\mathbf{t}$  in two specific distributions represented by  $\mathbf{M}$ , the task of the problem is to tell whether  $\mathbf{t}$  is in the span of  $\mathbf{M}$ .

**Our IBE in  $\text{NC}^1$ .** Our main technical contribution is a new approach of using the subset membership problem to transform an affine MAC to IBEs in the fine-grained setting. Our starting point is constructing a secure affine MAC in  $\text{NC}^1$ . We prove that, if the subset membership problem is hard in  $\text{NC}^1$ , then our MAC is secure for  $\text{NC}^1$  adversaries.

Next, we propose a generic construction of IBE based on affine MACs, following the BKP framework. In stark contrast to the BKP, our construction does not require pairings. Essentially, we develop a Groth-Sahai-like proof system in  $\text{NC}^1$  to prove the validity of our affine MAC. This proof system allows us to show that if our affine MAC is secure then our resulting IBE is secure in  $\text{NC}^1$ . At the core of our proof system is a new commitment scheme in  $\text{NC}^1$ , for which we achieve the hiding property by exploiting the concrete structure of matrices in  $D_0$ .

We give more details about the security proof. Firstly, the zero-knowledge property allows us to generate user secret keys for adversaries without knowing the MAC secret key. Secondly, we show that if an adversary can break the adaptive security of our IBE, then we can construct a reduction to break the security of our affine MAC. This is a crucial step, and we require some extractability of the proof system to extract the MAC forgery from the IBE adversary. In the BKP framework, this extractability can be achieved by computing the inversion of some matrix  $\mathbf{A} \in \mathbb{Z}_q^{k \times k}$  for some positive integer  $k$ . However, in our setting, inverting a matrix in  $\{0, 1\}^{n \times n}$  is impossible, otherwise, this will lead to a distinguisher for the subset membership problem in  $\text{NC}^1$ . Also, there is no known way to sample a matrix with its inverse efficiently [14]. To solve it, our proof system develop a new method in achieving this extractability without inverting any matrix. Our core idea is to prove that with a fresh random string  $r \leftarrow_{\$} \{0\} \times \{0, 1\}^{n-1}$ , it is possible to extract the forgery from our  $\text{NC}^1$ -commitments by switching the distribution of the public parameter  $\mathbf{A} \in D_0$  twice (from  $D_0$  to  $D_1$  and then back to  $D_0$ ) and changing the distribution of  $\mathbf{r}$  during the switching procedure.

**Dual system methodology in  $\text{NC}^1$  and ABE.** Our techniques for IBE can also be viewed as the dual system encryption methodology [31] in  $\text{NC}^1$ , which is an alternative interpretation of our approach. In our proof, there are two important technical steps, switching ciphertexts to invalid and randomizing MAC tags in the user secret keys. These correspond to switching ciphertexts and user secret keys from functional to semi-functional in the dual system encryption methodology [31,24,5,10]. Dual system methodology is very useful in constructing predicate encryption and it was only known with pairings. Our work is for the first time implementing the dual system methodology without pairings.

Similar to the extension from BKP-IBE [5] to CGW-ABE [10], we further extend our techniques in constructing ABEs. We first use predicate encodings [32,10] to generalize the notion of affine MAC and make it useful for constructing ABEs. After that, we upgrade our IBE techniques, and transform the generalized affine MAC to an adaptively secure ABE in  $\text{NC}^1$ .

**More extension and open problem.** We are optimistic that our approach can yield many more new public-key schemes in fine-grained cryptography. In particular, we show that our techniques can also be used to construct an efficient QA-NIZK in  $\text{NC}^1$  with adaptive soundness in the full paper. Roughly, we use the technique for proving the hiding property of the underlying commitment scheme in our IBE scheme to achieve adaptive soundness.

Also, we are optimistic that our approach can be used to construct hierarchical IBE [18,21]. We leave a detailed treatment of it as an open problem.

## 2 Preliminaries

**Notations.** We note that all arithmetic computations are over  $GF(2)$  in this work. Namely, all arithmetic computations are performed with a modulus of 2. We write  $a \stackrel{\$}{\leftarrow} \mathcal{A}(b)$  (respectively,  $a = \mathcal{A}(b)$ ) to denote the random variable outputted by a probabilistic (respectively, deterministic) algorithm  $\mathcal{A}$  on input  $b$ . By  $x \stackrel{\$}{\leftarrow} \mathcal{S}$  we denote the process of sampling an element  $x$  from a set or distribution  $\mathcal{S}$  uniformly at random. By  $\mathbf{x} \in \{0, 1\}^n$  we denote a column vector with size  $n$  and by, say,  $\mathbf{x} \in \{1\} \times \{0, 1\}^{n-1}$  we mean that the first element of  $\mathbf{x}$  is 1. By  $[n]$  we denote the set  $\{1, \dots, n\}$ . By  $x_i$  (respectively,  $x_i$ ) we denote the  $i$ th element of a vector  $\mathbf{x}$  (respectively,  $\mathbf{x}$ ). By  $\text{negl}$  we denote an unspecified negligible function.

For a matrix  $\mathbf{A} \in \{0, 1\}^{n \times t}$  with  $\text{rank } t' < n$ , we denote the sets  $\{\mathbf{y} | \exists \mathbf{x} \text{ s.t. } \mathbf{y} = \mathbf{A}\mathbf{x}\}$  and  $\{\mathbf{x} | \mathbf{A}\mathbf{x} = \mathbf{0}\}$  by  $\text{Im}(\mathbf{A})$  (i.e., the span of  $\mathbf{A}$ ) and  $\text{Ker}(\mathbf{A})$  respectively. By  $\mathbf{A}^\perp \in \{0, 1\}^{n \times (n-t')}$  we denote a matrix consisting of  $n - t'$  linear independent column vectors in the kernel of  $\mathbf{A}^\top$ . Note that for any  $\mathbf{y} \notin \text{Im}(\mathbf{A})$ , we have  $\mathbf{y}^\top \mathbf{A}^\perp \neq \mathbf{0}$ . By  $(a_{ij})_{i \in [l], j \in [m]}$  we denote the matrix  $\begin{pmatrix} a_{11} \cdots a_{1m} \\ \vdots \vdots \vdots \\ a_{l1} \cdots a_{lm} \end{pmatrix}$ . Let  $\mathbf{A} = (a_{ij})_{i \in [l], j \in [m]}$  be an  $l \times m$  matrix and  $\mathbf{B} = (\mathbf{B}_{ij})_{i \in [m], j \in [n]}$  be a large matrix consisting of  $m \times n$  matrices  $\mathbf{B}_{ij}$  for all  $i \in [m]$  and  $j \in [n]$ . By  $h \odot \mathbf{A}$  we denote  $(h \cdot a_{ij})_{i \in [l], j \in [m]}$  and by  $\mathbf{A} \odot \mathbf{B}$  we denote

$$\left( \sum_{k=1}^m a_{ik} \odot \mathbf{B}_{kj} \right)_{i \in [l], j \in [n]}.$$

By  $\mathbf{M}_0^n$ ,  $\mathbf{M}_1^n$ , and  $\mathbf{N}^n$ , we denote the following  $n \times n$  matrices:

$$\mathbf{M}_0^n = \begin{pmatrix} 0 & \cdots & 0 & 0 \\ 1 & 0 & & 0 \\ 0 & 1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{pmatrix}, \quad \mathbf{M}_1^n = \begin{pmatrix} 0 & \cdots & 0 & 1 \\ 1 & 0 & & 0 \\ 0 & 1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{pmatrix}, \quad \mathbf{N}^n = \begin{pmatrix} 0 & \cdots & & 0 \\ \vdots & 0 & \cdots & 0 \\ 0 & \ddots & \vdots & \\ 1 & 0 & \cdots & 0 \end{pmatrix},$$

and by  $\mathbf{0}$  we denote a zero vector  $(0, \dots, 0)^\top$ .

**Games.** We follow [5] to use code-based games for defining and proving security. A game  $\mathbf{G}$  contains procedures `INIT` and `FINALIZE`, and some additional procedures  $P_1, \dots, P_n$ , which are defined in pseudo-code. All variables in a game are initialized as 0, and all sets are empty (denote by  $\emptyset$ ). An adversary  $\mathcal{A} = \{a_\lambda\}_{\lambda \in \mathbb{N}}$  is executed in game  $\mathbf{G}$  w.r.t. the security parameter  $\lambda$  (denote by  $\mathbf{G}^{a_\lambda}$ ) if  $a_\lambda$  first calls `INIT`, obtaining its output. Next, it may make arbitrary queries to  $P_i$  (according to their specification) and obtain their output. Finally, it makes one single call to `FINALIZE`( $\cdot$ ) and stops. We use  $\mathbf{G}^{a_\lambda} \Rightarrow d$  to denote that  $\mathbf{G}$  outputs  $d$  after interacting with  $a_\lambda$ , and  $d$  is the output of `FINALIZE`.

## 2.1 Function Families

In this section, we recall the definitions of function families,  $\text{NC}^1$  circuits,  $\text{AC}^0[2]$  circuits, and  $\oplus\text{L}/\text{poly}$ . Note that  $\text{AC}^0[2] \subsetneq \text{NC}^1$  [27,30].

**Definition 1 (Function Family).** *A function family is a family of (possibly randomized) functions  $\mathcal{F} = \{f_\lambda\}_{\lambda \in \mathbb{N}}$ , where for each  $\lambda$ ,  $f_\lambda$  has a domain  $D_\lambda^f$  and a range  $R_\lambda^f$ .*

**Definition 2 ( $\text{NC}^1$ ).** *The class of (non-uniform)  $\text{NC}^1$  function families is the set of all function families  $\mathcal{F} = \{f_\lambda\}_{\lambda \in \mathbb{N}}$  for which there is a polynomial  $p(\cdot)$  and constant  $c$  such that for each  $\lambda$ ,  $f_\lambda$  can be computed by a (randomized) circuit of size  $p(\lambda)$ , depth  $c \log(\lambda)$ , and fan-in 2 using AND, OR, and NOT gates.*

**Definition 3 ( $\text{AC}^0[2]$ ).** *The class of (non-uniform)  $\text{AC}^0[2]$  function families is the set of all function families  $\mathcal{F} = \{f_\lambda\}_{\lambda \in \mathbb{N}}$  for which there is a polynomial  $p(\cdot)$  and constant  $c$  such that for each  $\lambda$ ,  $f_\lambda$  can be computed by a (randomized) circuit of size  $p(\lambda)$ , depth  $c$ , and unbounded fan-in using AND, OR, NOT, and PARITY gates.*

One can see that multiplication of a constant number of matrices can be performed in  $\text{AC}^0[2]$ , since it can be done in constant depth with PARITY gates.

**Definition 4 ( $\oplus\text{L}/\text{poly}$ ).**  *$\oplus\text{L}/\text{poly}$  is the set of all boolean function families  $\mathcal{F} = \{f_\lambda\}_{\lambda \in \mathbb{N}}$  for which there is a constant  $c$  such that for each  $\lambda$ , there is a non-deterministic Turing machine  $\mathcal{M}_\lambda$  such that for each input  $x$  with length  $\lambda$ ,  $\mathcal{M}_\lambda(x)$  uses at most  $c \log(\lambda)$  space, and  $f_\lambda(x)$  is equal to the parity of the number of accepting paths of  $\mathcal{M}_\lambda(x)$ .*

## 2.2 Sampling Procedure

We now recall the definitions of four sampling procedures LSamp, RSamp, ZeroSamp, and OneSamp in Figure 1. Note that the output of ZeroSamp( $n$ ) is always a matrix of rank  $n - 1$  and the output of OneSamp( $n$ ) is always a matrix of full rank [13].

We now recall several assumptions and lemmata on ZeroSamp and OneSamp given in [13].

**Definition 5 (Fine-grained matrix linear assumption [13]).** *There exists a polynomial  $n = n(\lambda)$  in the security parameter  $\lambda$  such that for any family  $\mathcal{A} = \{a_\lambda\}_{\lambda \in \mathbb{N}}$  in  $\text{NC}^1$ , we have*

$$\begin{aligned} & |\Pr[a_\lambda(\mathbf{M}) = 1 \mid \mathbf{M} \stackrel{\$}{\leftarrow} \text{ZeroSamp}(n)] - \\ & \Pr[a_\lambda(\mathbf{M}') = 1 \mid \mathbf{M}' \stackrel{\$}{\leftarrow} \text{OneSamp}(n)]| \leq \text{negl}(\lambda). \end{aligned}$$

**Lemma 1 (Lemma 4.3 in [13]).** *If  $\text{NC}^1 \subsetneq \oplus\text{L}/\text{poly}$ , then the fine-grained matrix linear assumption holds.*



<p><b>LSamp</b>(<math>n</math>):</p> <p>For all <math>i, j \in [n]</math> and <math>i &lt; j</math>:</p> $r_{i,j} \stackrel{\$}{\leftarrow} \{0, 1\}$ <p>Return</p> $\begin{pmatrix} 1 & r_{1,2} & \cdots & r_{1,n-1} & r_{1,n} \\ 0 & 1 & r_{2,3} & \cdots & r_{2,n} \\ 0 & 0 & \ddots & & \vdots \\ \vdots & \vdots & \ddots & 1 & r_{n-1,n} \\ 0 & \cdots & 0 & 0 & 1 \end{pmatrix}$	<p><b>RSamp</b>(<math>n</math>):</p> <p>For <math>i = 1, \dots, n-1</math></p> $r_i \stackrel{\$}{\leftarrow} \{0, 1\}$ <p>Return</p> $\begin{pmatrix} 1 & \cdots & 0 & r_1 \\ 0 & 1 & & r_2 \\ 0 & 0 & \ddots & \vdots \\ \vdots & \vdots & \ddots & 1 & r_{n-1} \\ 0 & \cdots & 0 & 0 & 1 \end{pmatrix}$	<p><b>ZeroSamp</b>(<math>n</math>):</p> $\mathbf{R}_0 \stackrel{\$}{\leftarrow} \text{LSamp}(n) \in \{0, 1\}^{n \times n}$ $\mathbf{R}_1 \stackrel{\$}{\leftarrow} \text{RSamp}(n) \in \{0, 1\}^{n \times n}$ <p>Return <math>\mathbf{R}_0 \mathbf{M}_0^n \mathbf{R}_1 \in \{0, 1\}^{n \times n}</math></p> <p><b>OneSamp</b>(<math>n</math>):</p> $\mathbf{R}_0 \stackrel{\$}{\leftarrow} \text{LSamp}(n)$ $\mathbf{R}_1 \stackrel{\$}{\leftarrow} \text{RSamp}(n)$ <p>Return <math>\mathbf{R}_0 \mathbf{M}_1^n \mathbf{R}_1 \in \{0, 1\}^{n \times n}</math></p>
---	--	--

**Fig. 1.** Definitions of LSamp, RSamp, ZeroSamp, and OneSamp.  $n = n(\lambda)$  is a polynomial in the security parameter  $\lambda$ .

**Remark.** Notice that for any polynomial  $n = n(\lambda)$ , we have  $\{f_n\}_{\lambda \in \mathbb{N}} \in \text{NC}^1$  iff  $\{f_\lambda\}_{\lambda \in \mathbb{N}} \in \text{NC}^1$  since  $O(\log(n(\lambda))) = O(\log(\lambda))$ . Hence, in the above lemma, we can also set  $n(\cdot)$  as an identity function, i.e.,  $n = \lambda$ . For simplicity, in the rest of the paper, we always let **ZeroSamp**( $\cdot$ ) and **OneSamp**( $\cdot$ ) take as input  $\lambda$ .

The following lemma implies that for a matrix  $\mathbf{M}^\top$  sampled by **ZeroSamp**( $\lambda$ ), there is a unique non-zero vector with the first (respectively, last) element being 1 in the kernel of  $\mathbf{M}$  (respectively,  $\mathbf{M}^\top$ ).

**Lemma 2 (Lemma 3 in [15]).** *For all  $\lambda \in \mathbb{N}$  and all  $\mathbf{M}^\top \in \text{ZeroSamp}(\lambda)$ , it holds that  $\text{Ker}(\mathbf{M}^\top) = \{\mathbf{0}, \mathbf{k}\}$  where  $\mathbf{k}$  is a vector such that  $\mathbf{k} \in \{0, 1\}^{\lambda-1} \times \{1\}$ .*

**Lemma 3 (Lemma 4 in [15]).** *For all  $\lambda \in \mathbb{N}$  and all  $\mathbf{M}^\top \in \text{ZeroSamp}(\lambda)$ , it holds that  $\text{Ker}(\mathbf{M}) = \{\mathbf{0}, \mathbf{k}\}$  where  $\mathbf{k}$  is a vector such that  $\mathbf{k} \in \{1\} \times \{0, 1\}^{\lambda-1}$ , i.e., there must exist  $\mathbf{M}^\perp \in \{1\} \times \{0, 1\}^{\lambda-1}$ .*

The following lemma indicates a simple relation between the distributions of the outputs of **ZeroSamp**( $\lambda$ ) and **OneSamp**( $\lambda$ ).

**Lemma 4 (Lemma 7 in [15]).** *For all  $\lambda \in \mathbb{N}$ , the distributions of  $\mathbf{M} + \mathbf{N}^\lambda$  and  $\mathbf{M}'$  are identical, where  $\mathbf{M}^\top \stackrel{\$}{\leftarrow} \text{ZeroSamp}(\lambda)$  and  $\mathbf{M}'^\top \stackrel{\$}{\leftarrow} \text{OneSamp}(\lambda)$ .*

We now give two lemmata showing that when sampling a random vector  $\mathbf{w}$  from  $\{0, 1\}^\lambda$ , the first element of  $\mathbf{w}$  does not affect the distribution of  $\mathbf{M}\mathbf{w}$  for  $\mathbf{M}^\top \in \text{ZeroSamp}(\lambda)$ .

**Lemma 5 (Lemma 5 in [15]).** *For all  $\lambda \in \mathbb{N}$  and all  $\mathbf{M}^\top \in \text{ZeroSamp}(\lambda)$ , it holds that*

$$\text{Im}(\mathbf{M}) = \{\mathbf{x} | \mathbf{w} \in \{0\} \times \{0, 1\}^{\lambda-1}, \mathbf{x} = \mathbf{M}\mathbf{w}\} = \{\mathbf{x} | \mathbf{w} \in \{1\} \times \{0, 1\}^{\lambda-1}, \mathbf{x} = \mathbf{M}\mathbf{w}\}.$$

**Lemma 6.** *For all  $\lambda \in \mathbb{N}$  and all  $\mathbf{M}^\top \in \text{ZeroSamp}(\lambda)$ , the distributions of  $\mathbf{x}$  and  $\mathbf{x}'$  are identical, where  $\mathbf{w} \stackrel{\$}{\leftarrow} \{0\} \times \{0, 1\}^{\lambda-1}$ ,  $\mathbf{w}' \stackrel{\$}{\leftarrow} \{1\} \times \{0, 1\}^{\lambda-1}$ ,  $\mathbf{x} = \mathbf{M}\mathbf{w}$ , and  $\mathbf{x}' = \mathbf{M}\mathbf{w}'$ .*



*Proof.* According to Lemma 3, for any  $\mathbf{M}^\top \in \text{ZeroSamp}(\lambda)$ , there exists  $\mathbf{k} \in \text{Ker}(\mathbf{M})$  such that  $\mathbf{k} \in \{1\} \times \{0, 1\}^{\lambda-1}$ . Therefore, the distributions of  $(\mathbf{w} + \mathbf{k})$ , where  $\mathbf{w} \leftarrow^{\$} \{0\} \times \{0, 1\}^{\lambda-1}$ , and  $\mathbf{w}' \leftarrow^{\$} \{1\} \times \{0, 1\}^{\lambda-1}$  are identical. Moreover, we have  $\mathbf{M}\mathbf{w} = \mathbf{M}(\mathbf{w} + \mathbf{k})$ . Hence, the distributions of  $\mathbf{M}\mathbf{w}$  and  $\mathbf{M}\mathbf{w}'$  are identical, completing the proof of Lemma 6.  $\square$

Below we recall the a theorem implicitly given in [15] as the subset membership problem for an HPS. Roughly, it shows that for  $\mathbf{M}^\top \leftarrow^{\$} \text{ZeroSamp}(\lambda)$ , a vector sampled from the span of  $\mathbf{M}$  is indistinguishable from one sampled outside the span of  $\mathbf{M}$  for any adversary in  $\text{NC}^1$ . We refer the reader to the full paper for the proof.

**Definition 6 (Fine-grained subset membership problem [15]).** Let  $\text{SY} = \{\text{SampYes}_\lambda\}_{\lambda \in \mathbb{N}}$  and  $\text{SN} = \{\text{SampNo}_\lambda\}_{\lambda \in \mathbb{N}}$  be function families described in Figure 2. For all  $\lambda \in \mathbb{N}$ , all  $\mathbf{M}^\top \in \text{ZeroSamp}(\lambda)$ , and all  $\mathbf{x} \in \text{SampNo}_\lambda(\mathbf{M})$ , we have  $\mathbf{x} \in \{0, 1\}^\lambda \setminus \text{Im}(\mathbf{M})$ , then for  $\mathbf{M}^\top \leftarrow^{\$} \text{ZeroSamp}(\lambda)$  and any adversary  $\mathcal{A} = \{a_\lambda\}_{\lambda \in \mathbb{N}} \in \text{NC}^1$ , we have

$$\left| \Pr[a_\lambda(\mathbf{x}) = 1 \mid \mathbf{x} \leftarrow^{\$} \text{SampYes}_\lambda(\mathbf{M})] - \Pr[a_\lambda(\mathbf{x}) = 1 \mid \mathbf{x} \leftarrow^{\$} \text{SampNo}_\lambda(\mathbf{M})] \right| \leq \text{negl}(\lambda).$$

$\text{SampYes}_\lambda(\mathbf{M} \in \{0, 1\}^{\lambda \times \lambda}):$ $\mathbf{w} \leftarrow^{\$} \{1\} \times \{0, 1\}^{\lambda-1}$ Return $\mathbf{x} = \mathbf{M}\mathbf{w}$	$\text{SampNo}_\lambda(\mathbf{M} \in \{0, 1\}^{\lambda \times \lambda}):$ $\mathbf{w} \leftarrow^{\$} \{1\} \times \{0, 1\}^{\lambda-1}$ Return $\mathbf{x} = (\mathbf{M} + \mathbf{N}^\lambda)\mathbf{w}$ .
---	---

**Fig. 2.** Definitions of SY and SN. Note that  $\text{SY}, \text{SN} \in \text{AC}^0[2]$ , since they only involve operations including sampling random bits and multiplication of a matrix and a vector.

**Theorem 1 ([15]).** If  $\text{NC}^1 \subsetneq \oplus\text{L}/\text{poly}$ , then the fine-grained subset membership problem (see Definition 6) holds.

**Remark.** Note that the subset membership problem in [15] gives a stronger result additionally showing that the output distributions of  $\text{SampYes}_\lambda(\mathbf{M})$  and  $\text{SampNo}_\lambda(\mathbf{M})$  are identical to the uniform distributions over  $\text{Im}(\mathbf{M})$  and  $\{0, 1\}^\lambda \setminus \text{Im}(\mathbf{M})$  respectively. We only need a weak form of it in this work.

### 2.3 Predicate Encodings

We now recall the definition of predicate encodings. As in [10], our resulting construction of ABE is generally based on a predicate encoding. By exploiting various types of encodings, we can achieve a broad class of ABEs.

Our definitions are slightly different from the original definition in [10], in that our definition is over  $GF(2)$  rather than  $GF(p)$ , and we require that the encodings are performed in a circuit class  $\mathcal{C}_1$ .

**Definition 7 (Predicate Encoding [10]).** Let  $P = \{p_\lambda\}_{\lambda \in \mathbb{N}}$  with  $p_\lambda : \mathcal{X} \times \mathcal{Y} \rightarrow \{0, 1\}$  be a predicate, where  $\mathcal{X}$  and  $\mathcal{Y}$  are polynomial-sized spaces associated with  $\lambda$ . An  $\mathcal{C}_1$ -predicate encoding for  $P$  is a function family  $PE = \{rE_\lambda, kE_\lambda, sE_\lambda, sD_\lambda, rD_\lambda\}_{\lambda \in \mathbb{N}} \in \mathcal{C}_1$  with

- $rE_\lambda : \mathcal{Y} \times \{0, 1\}^\ell \rightarrow \{0, 1\}^\eta$ ,
- $kE_\lambda : \mathcal{Y} \times \{0, 1\} \rightarrow \{0, 1\}^\eta$ ,
- $sE_\lambda : \mathcal{X} \times \{0, 1\}^\ell \rightarrow \{0, 1\}^\zeta$ ,
- $sD_\lambda : \mathcal{X} \times \mathcal{Y} \times \{0, 1\}^\zeta \rightarrow \{0, 1\}$ ,
- $rD_\lambda : \mathcal{X} \times \mathcal{Y} \times \{0, 1\}^\eta \rightarrow \{0, 1\}$ ,

where  $\ell = \ell(\lambda)$ ,  $\eta = \eta(\lambda)$ , and  $\zeta = \zeta(\lambda)$  are polynomials in  $\lambda$ .

Linearity is satisfied is for all  $\lambda \in \mathbb{N}$  and all  $(x, y) \in \mathcal{X} \times \mathcal{Y}$ ,  $rE_\lambda(y, \cdot)$ ,  $kE_\lambda(y, \cdot)$ ,  $sE_\lambda(x, \cdot)$ ,  $sD_\lambda(x, y, \cdot)$ , and  $rD_\lambda(x, y, \cdot)$  are  $\{0, 1\}$ -linear. Namely, for any  $y \in \mathcal{Y}$ , any  $\mathbf{w}_0, \mathbf{w}_1 \in \{0, 1\}^\ell$ , and any  $c \in \{0, 1\}$ , we have  $rE_\lambda(y, \mathbf{w}_0 + \mathbf{w}_1 \cdot c) = rE_\lambda(y, \mathbf{w}_0) + rE_\lambda(\mathbf{w}_1) \cdot c$ , and the same argument can be made for  $kE_\lambda$ ,  $sE_\lambda$ ,  $sD_\lambda$ , and  $rD_\lambda$ .

Restricted  $\alpha$ -reconstruction is satisfied if for all  $\lambda \in \mathbb{N}$ , all  $(x, y) \in \mathcal{X} \times \mathcal{Y}$  such that  $p_\lambda(x, y) = 1$ , all  $\mathbf{w} \in \{0, 1\}^\ell$ , and all  $\alpha \in \{0, 1\}$ , we have

$$rD_\lambda(x, y, rE_\lambda(y, \mathbf{w})) = sD_\lambda(x, y, sE_\lambda(x, \mathbf{w})) \text{ and } rD_\lambda(x, y, kE_\lambda(y, \alpha)) = \alpha.$$

$\alpha$ -privacy is satisfied if for all  $\lambda \in \mathbb{N}$ , all  $(x, y) \in \mathcal{X} \times \mathcal{Y}$  such that  $p_\lambda(x, y) = 0$ , and all  $\alpha \in \{0, 1\}$ , the following distributions are identical:

$$(x, y, \alpha, sE_\lambda(x, \mathbf{w}), rE_\lambda(y, \mathbf{w}) + kE_\lambda(y, \alpha)) \text{ and } (x, y, \alpha, sE_\lambda(x, \mathbf{w}), rE_\lambda(y, \mathbf{w})),$$

where  $\mathbf{w} \xleftarrow{\$} \{0, 1\}^\ell$ .

**Remark on notions for predicate encodings.** Similar to [10], we abuse the notion

$$rE_\lambda(x, \mathbf{W}) \text{ where } \mathbf{W} = (\mathbf{w}_{ij})_{i \in [l], j \in [m]} \text{ and } \mathbf{w}_{ij} \in \{0, 1\}^\ell$$

for all  $i, j$  to denote the matrix

$$(rE_\lambda(x, \mathbf{w}_{ij}))_{i \in [l], j \in [m]}.$$

The same argument is made for  $(kE_\lambda, sE_\lambda, sD_\lambda, rD_\lambda)$ .

**Encoding for equality.** We now give an example of predicate encoding  $PE_{\text{eq}}$  for equality  $P_{\text{eq}}$  in Figure 3. By instantiating our ABKEM given later in Section 5 with this encoding, we immediately achieve an IBKEM. Linearity is straightforward. Restricted  $\alpha$ -reconstruction follows from the fact that  $u + \mathbf{x}^\top \mathbf{w} = u + \mathbf{y}^\top \mathbf{w}$  when  $\mathbf{x} = \mathbf{y}$ , and  $\alpha$ -privacy follows from the fact that  $u + \mathbf{x}^\top \mathbf{w}$  and  $u + \mathbf{y}^\top \mathbf{w}$  are pairwise independent if  $\mathbf{x} \neq \mathbf{y}$ .

## 2.4 Attribute-based Key Encapsulation

We now give the definition of fine-grained ABKEM, the instantiation of which can be easily converted into ABEs by using a one-time symmetric cypher.

$\mathcal{X} = \{0, 1\}^n, \mathcal{Y} = \{0, 1\}^n$ $\ell = (1 + n), \eta = 1, \zeta = 1$	$\text{sE}_\lambda(\mathbf{x}, (u, \mathbf{w}^\top)^\top) = u + \mathbf{x}^\top \mathbf{w}$ $\text{rE}_\lambda(\mathbf{y}, (u, \mathbf{w}^\top)^\top) = u + \mathbf{y}^\top \mathbf{w}$ $\text{kE}_\lambda(\mathbf{y}, \alpha) = \alpha$ $\text{sD}_\lambda(\mathbf{x}, \mathbf{y}, c) = c$ $\text{rD}_\lambda(\mathbf{x}, \mathbf{y}, d) = d$
$\text{p}_\lambda(\mathbf{x}, \mathbf{y})$ : Return 1 iff $\mathbf{x} = \mathbf{y}$	

**Fig. 3.** Definitions of  $\text{P}_{\text{eq}} = \{\text{p}_\lambda\}_{\lambda \in \mathbb{N}}$  and  $\text{PE}_{\text{eq}} = \{\text{rE}_\lambda, \text{kE}_\lambda, \text{sE}_\lambda, \text{sD}_\lambda, \text{rD}_\lambda\}$ .

**Definition 8 (Attribute-based Key Encapsulation).** A  $\mathcal{C}_1$ -attribute-based key encapsulation (ABKEM) scheme for a predicate  $\text{P} = \{\text{p}_\lambda\}_\lambda$  is a function family  $\text{ABKEM} = \{\text{Gen}_\lambda, \text{USKGen}_\lambda, \text{Enc}_\lambda, \text{Dec}_\lambda\}_{\lambda \in \mathbb{N}} \in \mathcal{C}_1$  with the following properties.

- $\text{Gen}_\lambda$  returns the (master) public/secret key  $(\text{pk}, \text{sk})$ . We assume that  $\text{pk}$  implicitly defines value spaces  $\mathcal{X}$  and  $\mathcal{Y}$ , a key space  $\mathcal{K}$ , and a ciphertext space  $\mathcal{C}$ .
- $\text{USKGen}_\lambda(\text{sk}, \mathbf{y})$  returns a user secret-key  $\text{usk}[\mathbf{y}]$  for a value  $\mathbf{y} \in \mathcal{Y}$ .
- $\text{Enc}_\lambda(\text{pk}, \mathbf{x})$  returns a symmetric key  $\text{K} \in \mathcal{K}$  together with a ciphertext  $\text{ct} \in \mathcal{C}$  w.r.t.  $\mathbf{x} \in \mathcal{X}$ .
- $\text{Dec}_\lambda(\text{usk}[\mathbf{y}], \mathbf{x}, \text{ct})$  deterministically returns a decapsulated key  $\text{K} \in \mathcal{K}$  or the reject symbol  $\perp$ .

Perfect correctness is satisfied if for all  $\lambda \in \mathbb{N}$ , all  $(\text{pk}, \text{sk}) \in \text{Gen}_\lambda$ , all  $\mathbf{y} \in \mathcal{Y}$ , all  $\mathbf{x} \in \mathcal{X}$ , all  $\text{usk}[\mathbf{y}] \in \text{USKGen}_\lambda(\text{sk}, \mathbf{y})$ , and all  $(\text{K}, \text{ct}) \in \text{Enc}_\lambda(\text{pk}, \mathbf{x})$ , if  $\text{p}_\lambda(\mathbf{x}, \mathbf{y}) = 1$ , we have

$$\Pr[\text{Dec}_\lambda(\text{pk}, \text{usk}[\mathbf{y}], \text{ct}) = \text{K}] = 1.$$

The security requirement we consider is indistinguishability against chosen plaintext and attribute attacks (PR-AT-CPA) defined as follows.

**Definition 9 (PR-AT-CPA Security for ABKEM).** Let  $k(\cdot)$  and  $l(\cdot)$  be functions in  $\lambda$ . ABKEM is  $\mathcal{C}_2$ - $(k, l)$ -PR-AT-CPA secure if for any  $\mathcal{A} = \{a_\lambda\}_{\lambda \in \mathbb{N}} \in \mathcal{C}_2$ , where  $a_\lambda$  is allowed to make  $k$  rounds of adaptive queries to  $\text{USKGen}(\cdot)$  and each round it query  $l$  inputs, we have

$$|\Pr[\text{PR-AT-CPA}_{\text{real}}^{a_\lambda} \Rightarrow 1] - \Pr[\text{PR-AT-CPA}_{\text{rand}}^{a_\lambda} \Rightarrow 1]| \leq \text{negl}(\lambda),$$

where the experiments are defined in Figure 4.

### 3 Generalized Affine MAC

In this section, we give the definition of generalized affine MAC, which generalizes the notion of standard affine MAC [5] by using predicate encodings, and show how to construct it in the fine-grained setting under the assumption  $\text{NC}^1 \not\subseteq \oplus\text{L}/\text{poly}$ .

<p><u>INIT:</u>  <math>(pk, sk) \xleftarrow{\\$} \text{Gen}_\lambda</math>  Return <math>pk</math></p> <p><u>USKGEN(y):</u>  // <math>k(\lambda) \times l(\lambda)</math> queries  <math>\mathcal{Q}_y \xleftarrow{\\$} \mathcal{Q}_y \cup \{y\}</math>  Return <math>usk[id] \xleftarrow{\\$} \text{USKGen}_\lambda(sk, y)</math></p>	<p><u>ENC(x):</u>  // one query  <math>(K^*, ct^*) \xleftarrow{\\$} \text{Enc}_\lambda(pk, x)</math>  <div style="border: 1px solid black; padding: 2px; display: inline-block;"><math>K^* \xleftarrow{\\$} \mathcal{K}</math></div>  Return <math>(K^*, ct^*)</math></p> <p><u>FINALIZE(<math>\beta</math>):</u>  If <math>(p_\lambda(x, y) \neq 1</math> for all <math>y \in \mathcal{Q}_y</math>, return <math>\beta</math>  Else return 0</p>
--	---

**Fig. 4.** Security Games  $\text{PR-AT-CPA}_{\text{real}}$  and  $\text{PR-AT-CPA}_{\text{rand}}$  for defining PR-AT-CPA security for ABKEM. The boxed statement redefining  $K^*$  is only executed in game  $\text{PR-AT-CPA}_{\text{rand}}$ .

### 3.1 Definitions

The definition of generalized affine MAC is as follows.

**Definition 10 (Generalized Affine MAC).** Let  $\text{PE} = \{sE_\lambda, rE_\lambda, kE_\lambda, sD_\lambda, rD_\lambda\}_{\lambda \in \mathbb{N}} \in \mathcal{C}_1$  be a predicate encoding for  $\text{P} = \{p_\lambda\}_{\lambda \in \mathbb{N}}$ , where  $rE_\lambda : \mathcal{Y} \times \{0, 1\}^\ell \rightarrow \{0, 1\}^\eta$ ,  $kE_\lambda : \mathcal{X} \times \{0, 1\} \rightarrow \{0, 1\}^\eta$ , and  $sE_\lambda : \mathcal{X} \times \{0, 1\}^\ell \rightarrow \{0, 1\}^\zeta$ .

A  $\mathcal{C}_1$ -generalized affine message authentication code for PE is a function family  $\text{MAC}_{\text{GA}} = \{\text{Gen}_{\text{MAC}_\lambda}, \text{Tag}_\lambda, \text{Ver}_{\text{MAC}_\lambda}\}_{\lambda \in \mathbb{N}} \in \mathcal{C}_1$ .

1.  $\text{Gen}_{\text{MAC}_\lambda}$  returns  $sk_{\text{MAC}}$  containing  $(\mathbf{B}, \mathbf{X}, x')$ , where  $\mathbf{B} \in \text{ZeroSamp}(\lambda)$ ,  $\mathbf{X} \in \{0, 1\}^{\lambda \times \ell}$ , and  $x' \in \{0, 1\}$ .
2.  $\text{Tag}_\lambda(sk_{\text{MAC}}, m \in \mathcal{Y})$  returns a tag  $\tau = (\mathbf{t}, \mathbf{u}) \in \{0, 1\}^\lambda \times \{0, 1\}^\eta$ , computed as

$$\mathbf{t} \xleftarrow{\$} \text{SampYes}_\lambda(\mathbf{B}) \quad (1)$$

$$\mathbf{u} = rE_\lambda(m, \mathbf{X}^\top \mathbf{t}) + kE_\lambda(m, x') \in \{0, 1\}^\eta. \quad (2)$$

3.  $\text{Ver}_{\text{MAC}_\lambda}(sk_{\text{MAC}}, m, \tau = (\mathbf{t}, \mathbf{u}))$  verifies if equation (2) holds.

Correctness is satisfied if for any  $sk_{\text{MAC}} \in \text{Gen}_{\text{MAC}_\lambda}$ ,  $m \in \mathcal{Y}$ , and  $\tau \in \text{Tag}_\lambda(sk_{\text{MAC}}, m)$ , we have  $1 = \text{Ver}_{\text{MAC}_\lambda}(sk_{\text{MAC}}, m, \tau)$ .

The security requirement we consider is pseudorandomness against chosen message attacks (PR-CMA) defined as follows.

**Definition 11 (PR-CMA Security).** Let  $k = k(\lambda)$  and  $l = l(\lambda)$  be polynomials in  $\lambda$ .  $\text{MAC}_{\text{GA}}$  is  $\mathcal{C}_2$ - $(k, l)$ -PR-CMA secure if for any  $\mathcal{A} = \{a_\lambda\}_{\lambda \in \mathbb{N}} \in \mathcal{C}_2$ , where  $a_\lambda$  is allowed to make  $k$  rounds of adaptive queries to  $\text{EVAL}(\cdot)$  and each round it queries  $l$  inputs, we have

$$\Pr[\text{PR-CMA}_{\text{real}}^{a_\lambda} \Rightarrow 1] - \Pr[\text{PR-CMA}_{\text{rand}}^{a_\lambda} \Rightarrow 1] \leq \text{negl}(\lambda),$$

where the experiments are defined in Figure 5.

<p><b>INIT:</b>  <math>\text{sk}_{\text{MAC}} = (\mathbf{B}, \mathbf{X}, x') \xleftarrow{\\$} \text{Gen}_{\text{MAC}\lambda}(\text{par})</math>                  Return <math>\varepsilon</math></p> <p><b>EVAL(m):</b> // <math>k(\lambda) \times \ell(\lambda)</math> queries  <math>\mathcal{Q}_m = \mathcal{Q}_m \cup \{\mathbf{m}\}</math>                  Return <math>(\mathbf{t}, \mathbf{u}) \xleftarrow{\\$} \text{Tag}_\lambda(\text{sk}_{\text{MAC}}, \mathbf{m})</math></p>	<p><b>CHAL(m*):</b> //one query  <math>\mathbf{h}_0 = \text{sE}_\lambda(\mathbf{m}^*, \mathbf{X}^\top) \in \{0, 1\}^{\zeta \times \lambda}</math>  <math>h_1 = x' \in \{0, 1\}</math>  <math>h_1 \xleftarrow{\\$} \{0, 1\}</math></p> <p>Return <math>(h, \mathbf{h}_0, h_1)</math></p> <p><b>FINALIZE</b>(<math>\beta \in \{0, 1\}</math>):                  If <math>\text{p}_\lambda(\mathbf{m}^*, \mathbf{m}) \neq 1</math> for all <math>\mathbf{m} \in \mathcal{Q}_m</math>, return <math>\beta</math>                  Else return 0</p>
---	---

**Fig. 5.** Games  $\text{PR-CMA}_{\text{real}}$  and  $\text{PR-CMA}_{\text{rand}}$  for defining PR-CMA security. The boxed statement redefining  $h_1$  is only executed in game  $\text{PR-CMA}_{\text{rand}}$ .

Roughly, the PR-CMA security says that in the presence of many tags and a challenge token  $(h, \mathbf{h}_0, h_1)$ , an adversary cannot tell whether the  $h_1$  is honestly generated or randomness.

**Standard Affine MAC.** Let  $\mathbf{X} = (\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_n) \xleftarrow{\$} \{0, 1\}^{\lambda \times (n+1)}$ . When  $\text{p}_\lambda(\cdot)$  is an identity function,  $\mathbf{u}$  is computed as

$$u = \mathbf{x}_0^\top \mathbf{t} + \sum_{i=1}^n m_i \mathbf{x}_i^\top \mathbf{t} + x' \in \{0, 1\} \quad (3)$$

in Equation (2), and  $\mathbf{h}_0$  is computed as

$$\mathbf{h}_0 = h \cdot (\mathbf{x}_0^\top + \sum_{i=1}^n m_i \mathbf{x}_i^\top) \in \{0, 1\}^{1 \times \lambda} \quad (4)$$

in Figure 5, i.e., the predicate encoding is the one for equality (see Figure 3), the above definition become exactly the same as that of affine MAC given in [5] for the HPS based IBKEM, except that we only consider computations over  $GF(2)$  and  $\mathbf{t}$  is sampled by  $\text{SampYes}_\lambda$ . We give the definition as below.

**Definition 12 (Affine MAC [5]).** A Generalized affine MAC for the predicate  $\text{P}_{\text{eq}}$  and encoding  $\text{PE}_{\text{eq}}$  defined as in Figure 3 is said to be an affine MAC.

### 3.2 Construction

In this section, we give our construction of  $\text{AC}^0[2]$ -generalized affine MAC based on  $\text{NC}^1 \subsetneq \oplus\text{L}/\text{poly}$ . It is a natural extension of the standard affine MAC from an HPS in [5].

**Theorem 2.** If  $\text{NC}^1 \subsetneq \oplus\text{L}/\text{poly}$  and  $\text{PE} = \{\text{sE}_\lambda, \text{rE}_\lambda, \text{kE}_\lambda, \text{sD}_\lambda, \text{rD}_\lambda\}_{\lambda \in \mathbb{N}} \in \text{AC}^0[2]$  is a predicate encoding, where  $\text{rE}_\lambda : \mathcal{Y} \times \{0, 1\}^\ell \rightarrow \{0, 1\}^\eta$ ,  $\text{kE}_\lambda : \mathcal{Y} \times \{0, 1\} \rightarrow \{0, 1\}^\eta$ , and  $\text{sE}_\lambda : \mathcal{X} \times \{0, 1\}^\ell \rightarrow \{0, 1\}^\zeta$ , then  $\text{MAC}_{\text{GA}}$  is an  $\text{AC}^0[2]$ -generalized affine MAC that is  $\text{NC}^1$ - $(k, l)$ -PR-CMA secure, where  $k$  is any constant and  $l = l(\lambda)$  is any polynomial in  $\lambda$ .

$\text{Gen}_{\text{MAC}_\lambda}(\text{par}):$ $\mathbf{B}^\top \xleftarrow{\$} \text{ZeroSamp}(\lambda)$ $\mathbf{X} \xleftarrow{\$} \{0, 1\}^{\lambda \times \ell}$ $x' \xleftarrow{\$} \{0, 1\}$ Return $\text{sk}_{\text{MAC}} = (\mathbf{B}, \mathbf{X}, x')$	$\text{Tag}_\lambda(\text{sk}_{\text{MAC}}, m \in \mathcal{Y}):$ $\mathbf{t} \xleftarrow{\$} \text{SampYes}_\lambda(\mathbf{B})$ $\mathbf{u} = \text{rE}_\lambda(m, \mathbf{X}^\top \mathbf{t}) + \text{kE}_\lambda(m, x') \in \{0, 1\}^\eta$ Return $\tau = (\mathbf{t}, \mathbf{u})$  $\text{Ver}_{\text{MAC}_\lambda}(\text{sk}_{\text{MAC}}, m \in \mathcal{Y}, \tau):$ If $\mathbf{u} = \text{rE}_\lambda(m, \mathbf{X}^\top \mathbf{t}) + \text{kE}_\lambda(m, x')$ return 1 Else return 0
--	---

**Fig. 6.** Definition of  $\text{MAC}_{\text{GA}} = \{\text{Gen}_{\text{MAC}_\lambda}, \text{Tag}_\lambda, \text{Ver}_{\text{MAC}_\lambda}\}_{\lambda \in \mathbb{N}}$ .

*Proof.* First, we note that  $(\{\text{Gen}_{\text{MAC}_\lambda}\}_{\lambda \in \mathbb{N}}, \{\text{Tag}_\lambda\}_{\lambda \in \mathbb{N}}, \{\text{Ver}_{\text{MAC}_\lambda}\}_{\lambda \in \mathbb{N}})$  are computable in  $\text{AC}^0[2]$ , since they only involve operations including sampling random bits and multiplication of a constant number of matrices, which can be done in constant depth with PARITY gates. Also, it is straightforward that  $\text{MAC}_{\text{GA}}$  satisfies correctness.

We now prove that  $\text{MAC}_{\text{GA}}$  is  $\text{NC}^1$ - $(k, l)$ -PR-CMA secure by defining a sequence of intermediate games as in Figure 7.

Let  $\mathcal{A} = \{a_\lambda\}_{\lambda \in \mathbb{N}} \in \text{NC}^1$  be any adversary against the PR-CMA-security of  $\text{MAC}_{\text{GA}}$ . Game  $\text{G}_0$  is the real attack game. In games  $\text{G}_{1,i}$ , the first  $i - 1$  queries to the EVAL oracle are answered with  $(\mathbf{t}, \mathbf{u})$ , where  $\mathbf{t} \xleftarrow{\$} \text{SampNo}_\lambda(\mathbf{B})$  and  $\mathbf{u}$  contains no information on  $\text{kE}_\lambda(m, x')$ , and the remaining are answered as in the real scheme. To interpolate between  $\text{G}_{1,i}$  and  $\text{G}_{1,i+1}$ , we also define  $\text{G}'_{1,i}$ , which answers the  $i$ -th query to EVAL by picking  $\mathbf{t} \xleftarrow{\$} \text{SampNo}_\lambda(\mathbf{B})$ . By definition, we have  $\text{G}_0 = \text{G}_{1,1}$ .

**Lemma 7.**  $\Pr[\text{PR-CMA}_{\text{real}}^{a_\lambda} \Rightarrow 1] = \Pr[\text{G}_0^{a_\lambda} \Rightarrow 1] = \Pr[\text{G}'_{1,1}^{a_\lambda} \Rightarrow 1]$ .

**Lemma 8.** *There exists an adversary  $\mathcal{B}_{1,i} = \{b_\lambda^{1,i}\}_{\lambda \in \mathbb{N}} \in \text{NC}^1$  such that  $b_\lambda^{1,i}$  breaks the fine-grained subset membership problem (see Definition 6), which holds under  $\text{NC}^1 \not\subseteq \oplus \text{L/poly}$  according to Theorem 1, with probability*

$$|\Pr[\text{G}'_{1,i}^{a_\lambda} \Rightarrow 1] - \Pr[\text{G}_{1,i}^{a_\lambda} \Rightarrow 1]|.$$

*Proof.* Games  $\text{G}_{1,i}$  and  $\text{G}'_{1,i}$  only differ in the distribution of  $\mathbf{t}$  returned by the EVAL oracle for its  $i$ -th query. We build  $b_\lambda^{1,i}$  as follows.

The distinguisher  $b_\lambda^{1,i}$  runs in exactly the same way as the challenger in  $\text{G}_{1,i}$  except that for its  $i$ -th query, it obtains  $\mathbf{t}$  which is sampled as  $\mathbf{t} \xleftarrow{\$} \text{SampYes}_\lambda(\mathbf{B})$  or  $\mathbf{t} \xleftarrow{\$} \text{SampNo}_\lambda(\mathbf{B})$ . When  $a_\lambda$  outputs  $\beta \in \{0, 1\}$ ,  $b_\lambda$  outputs  $\beta$  if no  $m$  such that  $p_\lambda(m^*, m) = 1$  was queried to EVAL. Otherwise,  $b_\lambda$  outputs 0.

Since  $a_\lambda$  only makes constant rounds of queries, all the operations in  $b_\lambda$  are performed in  $\text{NC}^1$ . Hence, we have  $\mathcal{B}_{1,i} \in \text{NC}^1$ .

When  $\mathbf{t}$  is sampled as  $\mathbf{t} \xleftarrow{\$} \text{SampYes}_\lambda(\mathbf{B})$  (respectively,  $\mathbf{t} \xleftarrow{\$} \text{SampNo}_\lambda(\mathbf{B})$ ), the view of  $a_\lambda$  is exactly the same as its view in  $\text{G}_{1,i}$  (respectively,  $\text{G}'_{1,i}$ ). Thus the advantage of  $b_\lambda^{1,i}$  in breaking the subset membership problem is  $|\Pr[\text{G}'_{1,i}^{a_\lambda} \Rightarrow 1] - \Pr[\text{G}_{1,i}^{a_\lambda} \Rightarrow 1]|$ , completing this part of proof.  $\square$

<p><b>INIT:</b> // Games <math>G_0</math>-<math>G_2</math>  <math>\mathbf{B}^\top \xleftarrow{\\$} \text{ZeroSamp}(\lambda), x' \xleftarrow{\\$} \{0, 1\}</math>                  For <math>\mathbf{X} \xleftarrow{\\$} \{0, 1\}^{\lambda \times \ell}</math>                  Return <math>\varepsilon</math></p> <p><b>CHAL</b>(<math>m^* \in \mathcal{X}</math>): // Games <math>G_0</math>-<math>G_{1,Q+1}</math>, <math>G_2</math>  <math>\mathbf{h}_0 = \text{sE}_\lambda(m^*, \mathbf{X}^\top) \in \{0, 1\}^{\zeta \times \lambda}</math>  <math>h_1 = x' \in \{0, 1\}</math>  <math>h_1 \xleftarrow{\\$} \{0, 1\}</math>                  Return <math>(\mathbf{h}_0, h_1)</math></p> <p><b>FINALIZE</b>(<math>\beta \in \{0, 1\}</math>): // Games <math>G_0</math>-<math>G_2</math>                  If <math>p_\lambda(m^*, m) \neq 1</math> for all <math>m \in \mathcal{Q}_m</math>                      return <math>\beta</math>                  Else return 0</p> <p><b>EVAL</b>(<math>m</math>): // Game <math>G_2</math>  <math>\mathcal{Q}_m = \mathcal{Q}_m \cup \{m\}</math>  <math>\mathbf{t} \xleftarrow{\\$} \text{SampNo}_\lambda(\mathbf{B})</math>  <math>\mathbf{u} = \text{rE}_\lambda(m, \mathbf{X}^\top \mathbf{t}) \in \{0, 1\}^\eta</math>                  Return <math>(\mathbf{t}, \mathbf{u})</math></p>	<p><b>EVAL</b>(<math>m</math>): // Game <math>G_0</math>  <math>\mathcal{Q}_m = \mathcal{Q}_m \cup \{m\}</math>  <math>\mathbf{t} \xleftarrow{\\$} \text{SampYes}_\lambda(\mathbf{B})</math>  <math>\mathbf{u} = \text{rE}_\lambda(m, \mathbf{X}^\top \mathbf{t}) + \text{kE}_\lambda(m, x') \in \{0, 1\}^\eta</math>                  Return <math>(\mathbf{t}, \mathbf{u})</math></p> <p><b>EVAL</b>(<math>m</math>): // Games <math>G_{1,i}, G'_{1,i}</math>  <math>\mathcal{Q}_m = \mathcal{Q}_m \cup \{m\}</math> // Let <math>m</math> be the <math>c</math>-th query (<math>1 \leq c \leq k \cdot l</math>)                  If <math>c &lt; i</math> then                      <math>\mathbf{t} \xleftarrow{\\$} \text{SampNo}_\lambda(\mathbf{B})</math>                      <math>\mathbf{u} = \text{rE}_\lambda(m, \mathbf{X}^\top \mathbf{t}) \in \{0, 1\}^\eta</math>                  If <math>c &gt; i</math> then                      <math>\mathbf{t} \xleftarrow{\\$} \text{SampYes}_\lambda(\mathbf{B})</math>                      <math>\mathbf{u} = \text{rE}_\lambda(m, \mathbf{X}^\top \mathbf{t}) + \text{kE}_\lambda(m, x') \in \{0, 1\}^\eta</math>                  If <math>c = i</math> then                      <math>\mathbf{t} \xleftarrow{\\$} \text{SampYes}_\lambda(\mathbf{B})</math>                      <math>\mathbf{t} \xleftarrow{\\$} \text{SampNo}_\lambda(\mathbf{B})</math>                      <math>\mathbf{u} = \text{rE}_\lambda(m, \mathbf{X}^\top \mathbf{t}) + \text{kE}_\lambda(m, x') \in \{0, 1\}^\eta</math>                  Return <math>(\mathbf{t}, \mathbf{u})</math></p>
--	--

**Fig. 7.** Games  $G_0, (G_{1,i}, G'_{1,i})_{1 \leq i \leq k \cdot l}, G_{1,k \cdot l + 1}, G_2$  for the proof of Theorem 2.

**Lemma 9.**  $\Pr[G_{1,i+1}^{a_\lambda} \Rightarrow 1] = \Pr[G'_{1,i}^{a_\lambda} \Rightarrow 1]$ .

*Proof.* Let  $m$  be the  $i$ -th query to EVAL such that  $p_\lambda(m^*, m) \neq 1$  and let  $(\mathbf{t}, \mathbf{u})$  be its tag. We have  $\mathbf{t} \notin \text{Im}(\mathbf{B})$  due to Theorem 1. We use an information-theoretic argument to show that in  $G'_{1,i}$ ,  $\mathbf{u}$  does not reveal any information on  $x'$ . Information-theoretically,  $a_\lambda$  may learn  $\mathbf{B}^\top \mathbf{X}$  from each  $c$ -th query with  $c > i$ . Thus, for  $\mathbf{X} \xleftarrow{\$} \{0, 1\}^{\lambda \times \ell}$  and  $\mathbf{w} \xleftarrow{\$} \{0, 1\}^{\ell \times 1}$ ,  $a_\lambda$  information-theoretically obtains the distribution of

$$\begin{aligned}
 & \left( \begin{array}{c} \mathbf{X}^\top \mathbf{B} \\ \mathbf{h}_0 = h \odot \text{sE}_\lambda(m^*, \mathbf{X}^\top) \\ \mathbf{u} = \text{rE}_\lambda(m, \mathbf{X}^\top \mathbf{t}) + \text{kE}_\lambda(m, x') \end{array} \right) \\
 &= \left( \begin{array}{c} (\mathbf{X}^\top + \mathbf{wB}^{\perp \top}) \mathbf{B} \\ \mathbf{h}_0 = \text{sE}_\lambda(m^*, \mathbf{X}^\top + \mathbf{wB}^{\perp \top}) \\ \mathbf{u} = \text{rE}_\lambda(m, (\mathbf{X}^\top + \mathbf{wB}^{\perp \top}) \mathbf{t}) + \text{kE}_\lambda(m, x') \end{array} \right) \\
 &= \left( \begin{array}{c} \mathbf{X}^\top \mathbf{B} \\ \mathbf{h}_0 = \text{sE}_\lambda(m^*, \mathbf{X}^\top) + \text{sE}_\lambda(m^*, \mathbf{wB}^{\perp \top}) \\ \mathbf{u} = \text{rE}_\lambda(m, \mathbf{X}^\top \mathbf{t}) + \text{rE}_\lambda(m, \mathbf{w}) + \text{kE}_\lambda(m, x') \end{array} \right) (\because \mathbf{t} \notin \text{Im}(\mathbf{B})).
 \end{aligned}$$



This distribution is identical to the distribution of

$$\begin{pmatrix} \mathbf{X}^\top \mathbf{B} \\ \mathbf{h}_0 = \mathbf{sE}_\lambda(\mathbf{m}^*, \mathbf{X}^\top) + \mathbf{sE}_\lambda(\mathbf{m}^*, \mathbf{wB}^{\perp\top}) \\ \mathbf{u} = \mathbf{rE}_\lambda(\mathbf{m}, \mathbf{X}^\top \mathbf{t}) + \mathbf{rE}_\lambda(\mathbf{m}, \mathbf{w}) \end{pmatrix},$$

since the distribution of

$$(\mathbf{m}^*, \mathbf{m}, x', \mathbf{sE}_\lambda(\mathbf{m}^*, \mathbf{w}), \mathbf{rE}_\lambda(\mathbf{m}, \mathbf{w}) + \mathbf{kE}_\lambda(\mathbf{m}, x'))$$

and

$$(\mathbf{m}^*, \mathbf{m}, x', \mathbf{sE}_\lambda(\mathbf{m}^*, \mathbf{w}), \mathbf{rE}_\lambda(\mathbf{m}, \mathbf{w})),$$

are identical due to the  $\alpha$ -privacy of PE, completing this part of proof.  $\square$

**Lemma 10.**  $\Pr[\mathbf{G}_2^{a_\lambda} \Rightarrow 1] = \Pr[\mathbf{G}_{1,k,l+1}^{a_\lambda} \Rightarrow 1]$ .

*Proof.* Note that  $a_\lambda$  can ask at most  $k \cdot l$ -many EVAL queries. In both  $\mathbf{G}_{1,k,l+1}$  and  $\mathbf{G}_2$ , all the answers of EVAL are independent of  $x'$ . Hence,  $h_1$  from  $\mathbf{G}_{1,k,l+1}$  is uniform in the view of  $a_\lambda$ .  $\square$

We now do all the previous steps in the reverse order as in Figure 8. Then, by using the above arguments in a reverse order, we have the following lemma.

**Lemma 11.** *There exists an adversary  $\mathcal{B}^2 = \{b_\lambda^2\}_{\lambda \in \mathbb{N}} \in \text{NC}^1$  such that  $b_\lambda^2$  breaks the fine-grained subset membership problem with probability at least*

$$(|\Pr[\text{PR-CMA}_{\text{rand}}^{a_\lambda} \Rightarrow 1] - \Pr[\mathbf{G}_2^{a_\lambda} \Rightarrow 1]|)/(k \cdot l).$$

Putting all above together, Theorem 2 immediately follows.  $\square$

**An affine MAC.** By instantiating the underlying predicate encoding in Figure 6 with the encoding for equality (see Figure 3), we immediately obtain an affine MAC  $\text{MAC} = \{\text{Gen}_{\text{MAC}\lambda}, \text{Tag}_\lambda, \text{Ver}_{\text{MAC}\lambda}\}_{\lambda \in \mathbb{N}}$  as in Figure 9 for message space  $\{0, 1\}^\ell$ , which will be used to construct an IBE scheme in  $\text{NC}^1$  later. Formally, we have the following corollary derived from Theorem 2.

**Corollary 1.** *If  $\text{NC}^1 \subsetneq \oplus\text{L}/\text{poly}$ , then MAC is an  $\text{AC}^0[2]$ -affine MAC that is  $\text{NC}^1$ - $(k, l)$ -PR-CMA secure, where  $k$  is any constant and  $l = l(\lambda)$  is any polynomial in  $\lambda$ .*

## 4 Fine-grained Secure Identity-based Encryption

In this section, we present our fine-grained IBE scheme, which captures the core techniques of our ABE scheme given later in Section 5.

<p><u>INIT:</u> // Games <math>H_0</math>-<math>H_2</math>  <math>\mathbf{B}^\top \xleftarrow{\\$} \text{ZeroSamp}(\lambda); x' \xleftarrow{\\$} \{0, 1\}</math>  <math>\mathbf{X} \xleftarrow{\\$} \{0, 1\}^{\lambda \times \ell}</math>                  Return <math>\varepsilon</math></p> <p><u>CHAL(<math>m^*</math>):</u> // Games <math>H_0</math>-<math>H_2</math>  <math>\mathbf{h}_0 = \text{sE}_\lambda(m^*, \mathbf{X}^\top) \in \{0, 1\}^{\zeta \times \lambda}</math>  <math>h_1 \xleftarrow{\\$} \{0, 1\}</math>                  Return <math>(\mathbf{h}_0, h_1)</math></p> <p><u>FINALIZE(<math>\beta \in \{0, 1\}</math>):</u> // Games <math>H_0</math>-<math>H_2</math>                  If <math>\text{p}_\lambda(m^*, m) \neq 1</math> for all <math>y \in \mathcal{Q}_m</math>                      return <math>\beta</math>                  Else return 0</p> <p><u>EVAL(<math>m</math>):</u> // Game <math>H_0</math>  <math>\mathcal{Q}_m = \mathcal{Q}_m \cup \{m\}</math>  <math>\mathbf{t} \xleftarrow{\\$} \text{SampNo}_\lambda(\mathbf{B})</math>  <math>\mathbf{u} = \text{rE}_\lambda(m, \mathbf{X}^\top \mathbf{t}) \in \{0, 1\}^\eta</math>                  Return <math>(\mathbf{t}, \mathbf{u})</math></p>	<p><u>EVAL(<math>m</math>):</u> // Games <math>H_{1,i}, H'_{1,i}</math>  <math>\mathcal{Q}_m = \mathcal{Q}_m \cup \{m\}</math> // Let <math>m</math> be the <math>c</math>-th query (<math>1 \leq c \leq k \cdot l</math>)                  If <math>c &gt; i</math> then                      <math>\mathbf{t} \xleftarrow{\\$} \text{SampNo}_\lambda(\mathbf{B})</math>                      <math>\mathbf{u} = \text{rE}_\lambda(m, \mathbf{X}^\top \mathbf{t}) \in \{0, 1\}^\eta</math>                  If <math>c &lt; i</math> then                      <math>\mathbf{t} \xleftarrow{\\$} \text{SampYes}_\lambda(\mathbf{B})</math>                      <math>\mathbf{u} = \text{rE}_\lambda(m, \mathbf{X}^\top \mathbf{t}) + \text{kE}_\lambda(m, x') \in \{0, 1\}^\eta</math>                  If <math>c = i</math> then                      <math>\mathbf{t} \xleftarrow{\\$} \text{SampNo}_\lambda(\mathbf{B})</math>                      <math>\mathbf{t} \xleftarrow{\\$} \text{SampYes}_\lambda(\mathbf{B})</math>                      <math>\mathbf{u} = \text{rE}_\lambda(m) \mathbf{X}^\top \mathbf{t} + \text{kE}_\lambda(m, x') \in \{0, 1\}^\eta</math>                  Return <math>(\mathbf{t}, \mathbf{u})</math></p> <p><u>EVAL(<math>m</math>):</u> // Game <math>H_2</math>  <math>\mathcal{Q}_m = \mathcal{Q}_m \cup \{m\}</math>  <math>\mathbf{t} \xleftarrow{\\$} \text{SampYes}_\lambda(\mathbf{B})</math>  <math>\mathbf{u} = \text{rE}_\lambda(m, \mathbf{X}^\top \mathbf{t}) + \text{kE}_\lambda(m, x') \in \{0, 1\}^\eta</math>                  Return <math>(\mathbf{t}, \mathbf{u})</math></p>
---	---

Fig. 8. Games  $H_0, (H_{1,i}, H'_{1,i})_{1 \leq i \leq k \cdot l}, H_{1,k \cdot l + 1}, H_2$  for the proof of Lemma 11.

<p><u>Gen<math>_{\text{MAC}\lambda}(\text{par})</math>:</u>  <math>\mathbf{B}^\top \xleftarrow{\\$} \text{ZeroSamp}(\lambda)</math>  <math>\mathbf{x}_0, \dots, \mathbf{x}_\ell \xleftarrow{\\$} \{0, 1\}^\lambda</math>  <math>x' \xleftarrow{\\$} \{0, 1\}</math>                  Return <math>\text{sk}_{\text{MAC}} = (\mathbf{B}, \mathbf{x}_0, \dots, \mathbf{x}_\ell, x')</math></p>	<p><u>Tag<math>_\lambda(\text{sk}_{\text{MAC}}, m \in \{0, 1\}^\ell)</math>:</u>  <math>\mathbf{t} \xleftarrow{\\$} \text{SampYes}_\lambda(\mathbf{B})</math>  <math>u = (\mathbf{x}_0^\top + \sum_{i=1}^\ell m_i \cdot \mathbf{x}_i^\top) \mathbf{t} + x' \in \{0, 1\}</math>                  Return <math>\tau = (\mathbf{t}, u)</math></p> <p><u>Ver<math>_{\text{MAC}\lambda}(\text{sk}_{\text{MAC}}, \tau, m)</math>:</u>                  If <math>u = (\mathbf{x}_0^\top + \sum_{i=1}^\ell m_i \cdot \mathbf{x}_i^\top) \mathbf{t} + x'</math> return 1                  Else return 0</p>
--	--

Fig. 9. Definition of  $\text{MAC} = \{\text{Gen}_{\text{MAC}\lambda}, \text{Tag}_\lambda, \text{Ver}_{\text{MAC}\lambda}\}_{\lambda \in \mathbb{N}}$ .

#### 4.1 Definition

We now give the definition of fine-grained IBKEM, which is a special case of fine-grained ABKEM (see Definition 8) where the boolean predicate is restricted to be the equality predicate.

**Definition 13 (Identity-based Key Encapsulation).** A  $\mathcal{C}_1$ -identity key encapsulation (IBKEM) scheme is a function family  $\text{IBKEM} = \{\text{Gen}_\lambda, \text{USKGen}_\lambda, \text{Enc}_\lambda, \text{Dec}_\lambda\}_{\lambda \in \mathbb{N}} \in \mathcal{C}_1$  with the following properties.

- $\text{Gen}_\lambda$  returns the (master) public/secret key  $(\text{pk}, \text{sk})$ . We assume that  $\text{pk}$  implicitly defines an identity space  $\mathcal{ID}$ , a key space  $\mathcal{K}$ , and a ciphertext space  $\mathcal{C}$ .
- $\text{USKGen}_\lambda(\text{sk}, \text{id})$  returns a user secret-key  $\text{usk}[\text{id}]$  for an identity  $\text{id} \in \mathcal{ID}$ .
- $\text{Enc}_\lambda(\text{pk}, \text{id})$  returns a symmetric key  $K \in \mathcal{K}$  together with a ciphertext  $\text{ct} \in \mathcal{C}$  w.r.t.  $\text{id} \in \mathcal{ID}$ .
- $\text{Dec}_\lambda(\text{usk}[\text{id}], \text{id}, \text{ct})$  deterministically returns a decapsulated key  $K \in \mathcal{K}$  or the reject symbol  $\perp$ .

Perfect correctness is satisfied if for all  $\lambda \in \mathbb{N}$ , all  $(\text{pk}, \text{sk}) \in \text{Gen}_\lambda$ , all  $\text{id} \in \mathcal{ID}$ , all  $\text{usk}[\text{id}] \in \text{USKGen}_\lambda(\text{sk}, \text{id})$ , and all  $(K, \text{ct}) \in \text{Enc}_\lambda(\text{pk}, \text{id})$ , we have

$$\Pr[\text{Dec}_\lambda(\text{pk}, \text{usk}[\text{id}], \text{ct}) = K] = 1.$$

The security requirement we consider is indistinguishability against chosen plaintext and identity attacks (PR-ID-CPA) defined as follows.

**Definition 14 (PR-ID-CPA Security for IBKEM).** Let  $k(\cdot)$  and  $l(\cdot)$  be functions in  $\lambda$ . IBKEM is  $\mathcal{C}_2$ - $(k, l)$ -PR-ID-CPA secure if for any  $\mathcal{A} = \{a_\lambda\}_{\lambda \in \mathbb{N}} \in \mathcal{C}_2$ , where  $a_\lambda$  is allowed to make  $k$  rounds of adaptive queries to  $\text{USKGen}(\cdot)$  and each round it query  $l$  inputs, we have

$$|\Pr[\text{PR-ID-CPA}_{\text{real}}^{a_\lambda} \Rightarrow 1] - \Pr[\text{PR-ID-CPA}_{\text{rand}}^{a_\lambda} \Rightarrow 1]| \leq \text{negl}(\lambda),$$

where the experiments are defined in Figure 10.

<p><b>Procedure INIT:</b>  <math>(\text{pk}, \text{sk}) \stackrel{\\$}{\leftarrow} \text{Gen}_\lambda</math>  Return <math>\text{pk}</math></p> <p><b>Procedure USKGEN(id):</b>  //<math>k(\lambda) \times l(\lambda)</math> queries  <math>\mathcal{Q}_{\text{id}} \stackrel{\\$}{\leftarrow} \mathcal{Q}_{\text{id}} \cup \{\text{id}\}</math>  Return <math>\text{usk}[\text{id}] \stackrel{\\$}{\leftarrow} \text{USKGen}_\lambda(\text{sk}, \text{id})</math></p>	<p><b>Procedure ENC(id*):</b>  //one query  <math>(K^*, \text{ct}^*) \stackrel{\\$}{\leftarrow} \text{Enc}_\lambda(\text{pk}, \text{id}^*)</math>  <div style="border: 1px solid black; display: inline-block; padding: 2px;"><math>K^* \stackrel{\\$}{\leftarrow} \mathcal{K}</math></div>  Return <math>(K^*, \text{ct}^*)</math></p> <p><b>Procedure FINALIZE(<math>\beta</math>):</b>  Return <math>(\text{id}^* \notin \mathcal{Q}_{\text{id}}) \wedge \beta</math></p>
--	--

**Fig. 10.** Security Games  $\text{PR-ID-CPA}_{\text{real}}$  and  $\text{PR-ID-CPA}_{\text{rand}}$  for defining PR-ID-CPA-security for IBKEM. The boxed statement redefining  $K^*$  is only executed in game  $\text{PR-ID-CPA}_{\text{rand}}$ .

## 4.2 Construction

Let  $\text{MAC} = \{\text{Gen}_{\text{MAC}\lambda}, \text{Tag}_\lambda, \text{Ver}_{\text{MAC}\lambda}\}_{\lambda \in \mathbb{N}} \in \text{NC}^1$  be an affine MAC over  $\{0, 1\}^\lambda$  with message space  $\mathcal{ID}$  in Figure 9. Our IBKEM  $\text{IBKEM} = \{\text{Gen}_\lambda, \text{USKGen}_\lambda,$

$\text{Enc}_\lambda, \text{Dec}_\lambda\}_{\lambda \in \mathbb{N}}$  for key-space  $\mathcal{K} = \{0, 1\}$  and identity space  $\{0, 1\}^\ell$  is defined as in Figure 11.<sup>6</sup>

<p><b>Gen<sub>λ</sub>:</b>  <math>\mathbf{A}^\top \xleftarrow{\\$} \text{ZeroSamp}(\lambda)</math>  <math>\text{sk}_{\text{MAC}} = (\mathbf{B}, \mathbf{x}_0, \dots, \mathbf{x}_\ell, x') \xleftarrow{\\$} \text{Gen}_{\text{MAC}\lambda}(\text{par})</math>                  For <math>i = 0, \dots, \ell</math>:  <math>\mathbf{Y}_i \xleftarrow{\\$} \{0, 1\}^{(\lambda-1) \times \lambda}</math>  <math>\mathbf{Z}_i = (\mathbf{Y}_i^\top \parallel \mathbf{x}_i) \mathbf{A} \in \{0, 1\}^{\lambda \times \lambda}</math>  <math>\mathbf{y}' \xleftarrow{\\$} \{0, 1\}^{\lambda-1}</math>  <math>\mathbf{z}' = (\mathbf{y}'^\top \parallel x') \mathbf{A} \in \{0, 1\}^{1 \times \lambda}</math>  <math>\text{pk} = (\mathbf{A}, (\mathbf{Z}_i)_{0 \leq i \leq \ell}, \mathbf{z}')</math>  <math>\text{sk} = (\text{sk}_{\text{MAC}}, (\mathbf{Y}_i)_{0 \leq i \leq \ell}, \mathbf{y}')</math>                  Return <math>(\text{pk}, \text{sk})</math></p> <p><b>USKGen<sub>λ</sub>(sk, id ∈ {0, 1}<sup>ℓ</sup>):</b>  <math>(\mathbf{t}, u) \xleftarrow{\\$} \text{Tag}_\lambda(\text{sk}_{\text{MAC}}, \text{id})</math>  <math>\mathbf{v} = \mathbf{t}^\top (\mathbf{Y}_0^\top + \sum_{i=1}^\ell \text{id}_i \odot \mathbf{Y}_i^\top) + \mathbf{y}'^\top \in \{0, 1\}^{1 \times (\lambda-1)}</math>                  Return <math>\text{usk}[\text{id}] = (\mathbf{t}, u, \mathbf{v})</math></p>	<p><b>Enc<sub>λ</sub>(pk, id):</b>  <math>\mathbf{r} \xleftarrow{\\$} \{0\} \times \{0, 1\}^{\lambda-1}</math>  <math>\mathbf{c}_0 = \mathbf{A} \mathbf{r} \in \{0, 1\}^\lambda</math>  <math>\mathbf{c}_1 = (\mathbf{Z}_0 + \sum_{i=1}^\ell \text{id}_i \odot \mathbf{Z}_i) \mathbf{r} \in \{0, 1\}^\lambda</math>  <math>\mathbf{K} = \mathbf{z}' \cdot \mathbf{r} \in \{0, 1\}</math>.                  Return <math>\mathbf{K}</math> and <math>\text{ct} = (\mathbf{c}_0, \mathbf{c}_1)</math></p> <p><b>Dec<sub>λ</sub>(usk[id], id, ct):</b>                  Parse <math>\text{usk}[\text{id}] = (\mathbf{t}, u, \mathbf{v})</math>                  Parse <math>\text{ct} = (\mathbf{c}_0, \mathbf{c}_1) \in \{0, 1\}^\lambda \times \{0, 1\}^\lambda</math>  <math>\mathbf{K} = (\mathbf{v} \mid u) \mathbf{c}_0 - \mathbf{t}^\top \mathbf{c}_1</math>                  Return <math>\mathbf{K}</math></p>
---	--

**Fig. 11.** Definition of our IBKEM =  $\{\text{Gen}_\lambda, \text{USKGen}_\lambda, \text{Enc}_\lambda, \text{Dec}_\lambda\}_{\lambda \in \mathbb{N}}$  with identity space  $\{0, 1\}^\ell$  and key space  $\{0, 1\}$ .  $\text{id}_i$  denotes the  $i$ th bit of  $\text{id}$  for all  $i \in [\ell]$ .

**Theorem 3.** *Under the assumption  $\text{NC}^1 \not\subseteq \oplus\text{L}/\text{poly}$  and the  $\text{NC}^1$ -( $k, l$ )-PR-CMA security of MAC, where  $k$  is any constant and  $l = l(\lambda)$  is any polynomial in  $\lambda$ , IBKEM is an  $\text{AC}^0[2]$ -IBKEM that is  $\text{NC}^1$ -( $k, l$ )-PR-ID-CPA secure against  $\text{NC}^1$ .*

Due to the page limit, we refer the reader to the full paper for the proof of Theorem 3.

**Extension to IBKEM with large key space.** The key space of the above IBKEM is  $\{0, 1\}$ , while by running it in parallel, we can easily extend it to an IBKEM with large key space. The resulting scheme can still be performed in  $\text{AC}^0[2]$  since running in parallel does not increase the circuit depth. The same extension can be also made for our fine-grained secure ABKEM given later in Section 5.

**Extension to QA-NIZK.** Our techniques for proving the hiding property of the underlying commitment scheme in our IBKEM can also be used to construct an efficient fine-grained QA-NIZK in  $\text{NC}^1$  with adaptive soundness. We refer the reader to the full paper for details.

<sup>6</sup> The IBKEM can be straightforwardly extended to one with large key space as we will discuss later in this section.

## 5 Fine-grained Secure Attribute-Based Encryption

In this section, we generalize our IBE scheme as a fine-grained ABE scheme by using predicate encodings [32,10]. By instantiating the underlying encodings in different ways, we can achieve ABEs for inner product, non-zero inner product, spatial encryption, doubly spatial encryption, boolean span programs, and arithmetic span programs, and also broadcast encryption and fuzzy IBE schemes, which are computable in  $\text{AC}^0[2]$  and secure against  $\text{NC}^1 \subsetneq \oplus\text{L}/\text{poly}$ . We refer the reader to the full paper for several instances of the encodings and also to [10] for more instances. We note that the encodings in [10] are defined over  $GF(p)$ , while the ours are over  $GF(2)$ . However, the proofs for encodings in [10] can be adopted in our case, since the linearity and  $\alpha$ -reconstruction properties hold in  $GF(p)$  also hold in  $GF(2)$  and by the standard linear-independence arguments in  $GF(2)$ , the  $\alpha$ -privacy also holds in our case.

Let  $\text{PE} = \{\text{rE}_\lambda, \text{kE}_\lambda, \text{sE}_\lambda, \text{sD}_\lambda, \text{rD}_\lambda\}_{\lambda \in \mathbb{N}} \in \text{AC}^0[2]$  be a predicate encoding for  $\text{P} = \{\text{p}_\lambda\}_{\lambda \in \mathbb{N}}$  with  $\text{rE}_\lambda : \mathcal{Y} \times \{0,1\}^\ell \rightarrow \{0,1\}^\eta$ ,  $\text{kE}_\lambda : \mathcal{Y} \times \{0,1\} \rightarrow \{0,1\}^\eta$ ,  $\text{sE}_\lambda : \mathcal{X} \times \{0,1\}^\ell \rightarrow \{0,1\}^\zeta$ ,  $\text{sD}_\lambda : \mathcal{X} \times \mathcal{Y} \times \{0,1\}^\zeta \rightarrow \{0,1\}$ , and  $\text{rD}_\lambda : \mathcal{X} \times \mathcal{Y} \times \{0,1\}^\eta \rightarrow \{0,1\}$ . Let  $\text{MAC}_{\text{GA}} = \{\text{Gen}_{\text{MAC}_\lambda}, \text{Tag}_\lambda, \text{Ver}_{\text{MAC}_\lambda}\}_{\lambda \in \mathbb{N}} \in \text{AC}^0[2]$  be a PE-generalized affine MAC over  $\{0,1\}^\lambda$  with message space  $\mathcal{Y}$ . Our ABKEM  $\text{ABKEM} = \{\text{Gen}_\lambda, \text{USKGen}_\lambda, \text{Enc}_\lambda, \text{Dec}_\lambda\}_{\lambda \in \mathbb{N}}$  is defined as in Figure 12.

<p><b>Gen<sub>λ</sub>:</b>  <math>\mathbf{A}^\top \xleftarrow{\\$} \text{ZeroSamp}(\lambda)</math>  <math>\text{sk}_{\text{MAC}} = (\mathbf{B}, \mathbf{X}, x') \xleftarrow{\\$} \text{Gen}_{\text{MAC}_\lambda}(\text{par})</math>            For <math>\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_\ell)</math> and <math>i = 1, \dots, \ell</math>:  <math>\mathbf{Y}_i \xleftarrow{\\$} \{0,1\}^{(\lambda-1) \times \lambda}</math>  <math>\mathbf{Z}_i = (\mathbf{Y}_i^\top \parallel \mathbf{x}_i) \mathbf{A} \in \{0,1\}^{\lambda \times \lambda}</math>  <math>\mathbf{y}' \xleftarrow{\\$} \{0,1\}^{(\lambda-1)}</math>  <math>\mathbf{z}' = (\mathbf{y}'^\top \parallel x') \mathbf{A} \in \{0,1\}^{1 \times \lambda}</math>  <math>\text{pk} = (\mathbf{A}, (\mathbf{Z}_i)_{1 \leq i \leq \ell}, \mathbf{z}')</math>  <math>\text{sk} = (\text{sk}_{\text{MAC}}, (\mathbf{Y}_i)_{1 \leq i \leq \ell}, \mathbf{y}')</math>            Return <math>(\text{pk}, \text{sk})</math></p> <p><b>USKGen<sub>λ</sub>(sk, y ∈ Y):</b>  <math>(\mathbf{t}, \mathbf{u}) \xleftarrow{\\$} \text{Tag}_\lambda(\text{sk}_{\text{MAC}}, y)</math>  <math>\mathbf{v} = \text{rE}_\lambda(y, \begin{pmatrix} \mathbf{t}^\top \mathbf{Y}_1^\top \\ \vdots \\ \mathbf{t}^\top \mathbf{Y}_\ell^\top \end{pmatrix})</math>  <math>+ \text{kE}_\lambda(y, \mathbf{y}'^\top) \in \{0,1\}^{\eta \times (\lambda-1)}</math>            Return <math>\text{usk}[y] = (\mathbf{t}, \mathbf{u}, \mathbf{v})</math></p>	<p><b>Enc<sub>λ</sub>(pk, x ∈ X):</b>  <math>\mathbf{r} \xleftarrow{\\$} \{0\} \times \{0,1\}^{\lambda-1}</math>  <math>\mathbf{c}_0 = \mathbf{A} \mathbf{r} \in \{0,1\}^\lambda</math>  <math>\mathbf{C}_1 = \text{sE}_\lambda(x, \begin{pmatrix} \mathbf{r}^\top \mathbf{Z}_1^\top \\ \vdots \\ \mathbf{r}^\top \mathbf{Z}_\ell^\top \end{pmatrix}) \in \{0,1\}^{\zeta \times \lambda}</math>  <math>\mathbf{K} = \mathbf{z}' \cdot \mathbf{r} \in \{0,1\}</math>            Return <math>\mathbf{K}</math> and <math>\text{ct} = (\mathbf{c}_0, \mathbf{C}_1)</math></p> <p><b>Dec<sub>λ</sub>(pk, usk[y], ct):</b>            Parse <math>\text{usk}[y] = (\mathbf{t}, \mathbf{u}, \mathbf{v})</math>            Parse <math>\text{ct} = (\mathbf{c}_0, \mathbf{C}_1)</math>  <math>\mathbf{K} = \text{rD}_\lambda(x, y, \mathbf{v} \parallel \mathbf{u}) \mathbf{c}_0</math>  <math>- \text{sD}_\lambda(x, y, \mathbf{C}_1 \mathbf{t}) \in \{0,1\}</math>            Return <math>\mathbf{K}</math></p>
---	--

Fig. 12. Construction of  $\text{ABKEM} = \{\text{Gen}_\lambda, \text{USKGen}_\lambda, \text{Enc}_\lambda, \text{Dec}_\lambda\}_{\lambda \in \mathbb{N}}$ .

**Theorem 4.** *Under the assumption  $\text{NC}^1 \subsetneq \oplus\text{L}/\text{poly}$  and the  $\text{NC}^1\text{-}(k, l)\text{-sE}_\lambda\text{-PR-CMA}$ -security of  $\text{MAC}_{\text{GA}}$ , where  $k$  is any constant and  $l = l(\lambda)$  is any polynomial in  $\lambda$ ,  $\text{ABKEM}$  is an  $\text{AC}^0[2]\text{-ABKEM}$  that is  $\text{NC}^1\text{-}(k, l)\text{-PR-AT-CPA}$  secure against  $\text{NC}^1$ .*

*Proof.* First, we note that  $\{\text{Gen}_\lambda\}_{\lambda \in \mathbb{N}}$ ,  $\{\text{USKGen}_\lambda\}_{\lambda \in \mathbb{N}}$ ,  $\{\text{Enc}_\lambda\}_{\lambda \in \mathbb{N}}$ , and  $\{\text{Dec}_\lambda\}_{\lambda \in \mathbb{N}}$  are computable in  $\text{AC}^0[2]$ , since they only involve operations including multiplication of a constant number of matrices, sampling random bits, and running  $\text{MAC}_{\text{GA}} \in \text{AC}^0[2]$ .

By Equation (2) in Section 3.1, we have

$$\begin{aligned} & \text{rD}_\lambda(x, y, \mathbf{v} \parallel \mathbf{u}) \mathbf{c}_0 \\ = & \text{rD}_\lambda(x, y, \text{rE}_\lambda \left( y, \begin{pmatrix} \mathbf{t}^\top \mathbf{Y}_1^\top \\ \vdots \\ \mathbf{t}^\top \mathbf{Y}_\ell^\top \end{pmatrix} + \text{kE}_\lambda(y, \mathbf{y}'^\top) \parallel \begin{pmatrix} \mathbf{t}^\top \mathbf{x}_1 \\ \vdots \\ \mathbf{t}^\top \mathbf{x}_\ell \end{pmatrix} + \text{kE}_\lambda(y, x') \right) \mathbf{A} \mathbf{r} \end{aligned}$$

and

$$\text{sD}_\lambda(x, y, \mathbf{C}_1 \mathbf{t}) = \text{sD}_\lambda(x, y, \text{sE}_\lambda \left( x, \begin{pmatrix} \mathbf{t}^\top (\mathbf{Y}_1^\top \parallel \mathbf{x}_1) \\ \vdots \\ \mathbf{t}^\top (\mathbf{Y}_\ell^\top \parallel \mathbf{x}_\ell) \end{pmatrix} \right) \mathbf{A} \mathbf{r}.$$

Then, due to restricted  $\alpha$ -reconstruction (see Definition 7), the difference of the above equations yields  $\mathbf{K} = (\mathbf{y}'^\top \parallel x') \mathbf{A} \mathbf{r} = \mathbf{z}' \cdot \mathbf{r}$ , i.e., correctness is satisfied.

Let  $\mathcal{A} = \{a_\lambda\}_{\lambda \in \mathbb{N}}$  be any adversary against the  $\text{NC}^1\text{-}(k, l)\text{-PR-AT-CPA}$  security of  $\text{ABKEM}$ . We now prove the  $\text{NC}^1\text{-}(k, l)\text{-PR-AT-CPA}$  security by defining a sequence of games  $\text{G}_0\text{-G}_6$  as in Figure 13. Roughly, in the first four games, we show how to extract a challenge token for  $\text{MAC}_{\text{GA}}$  from the challenge session key and ciphertext by switching the distribution of  $\mathbf{A}$  twice and changing the distribution of the randomness  $\mathbf{r}$  during the switching procedure. In the last two games, we show that the commitments  $\mathbf{Z}_i$  and  $\mathbf{z}'$  perfectly hide the secrets, and the answers of queries made by  $a_\lambda$  reveal no useful information other than the tags and token for  $\text{MAC}$ .

**Lemma 12.**  $\Pr[\text{PR-AT-CPA}_{\text{real}}^{a_\lambda} \Rightarrow 1] = \Pr[\text{G}_1^{a_\lambda} \Rightarrow 1] = \Pr[\text{G}_0^{a_\lambda} \Rightarrow 1]$ .

*Proof.*  $\text{G}_0$  is the real attack game. In game  $\text{G}_1$ , we change the simulation of  $\mathbf{c}_0^*$ ,  $\mathbf{C}_1^*$  and  $\mathbf{K}^*$  in  $\text{ENC}(x)$  by substituting  $\mathbf{Z}_i$  and  $\mathbf{z}'$  with their respective definitions and substituting  $\mathbf{A}$  with  $\mathbf{A} + \mathbf{N}^\lambda$ . Since we have

$$\mathbf{N}^{\lambda, \mathbf{r}} = \begin{pmatrix} 0 & \cdots & 0 \\ \vdots & 0 & \cdots & 0 \\ 0 & \ddots & \vdots \\ 1 & 0 & \cdots & 0 \end{pmatrix} \begin{pmatrix} 0 \\ r_2 \\ \vdots \\ r_\lambda \end{pmatrix} = 0,$$

the view of  $a_\lambda$  in  $\text{G}_1$  is identical to its view in  $\text{G}_0$ , completing this part of proof.  $\square$

INIT: //Games  $G_0$ - $G_1$ ,  $G_2$ - $G_3$ ,  $G_4$ ,  $G_5$ - $G_6$

$\mathbf{A}^\top \xleftarrow{\$} \text{ZeroSamp}(\lambda)$ ,  $\mathbf{A}^\top \xleftarrow{\$} \text{OneSamp}(\lambda)$ ,  $\mathbf{A}^\top \xleftarrow{\$} \text{ZeroSamp}(\lambda)$

$\mathbf{R}_1 = \begin{pmatrix} \mathbf{I}_{\lambda-1} & \mathbf{0} \\ \tilde{\mathbf{r}}^\top & 1 \end{pmatrix}^\top \xleftarrow{\$} \text{RSamp}(\lambda)$ ,  $\mathbf{R}_0 \xleftarrow{\$} \text{LSamp}(\lambda)$ ,  $\mathbf{A}^\top = \mathbf{R}_0 \mathbf{M}_0^\lambda \mathbf{R}_1$

$\text{sk}_{\text{MAC}} = (\mathbf{B}, \mathbf{X}, x') \xleftarrow{\$} \text{Gen}_{\text{MAC}\lambda}(\mathcal{G})$

For  $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_\ell)$  and  $i = 1, \dots, \ell$ :

$\mathbf{Y}_i \xleftarrow{\$} \{0, 1\}^{(\lambda-1) \times \lambda}$ ,  $\mathbf{Z}_i = (\mathbf{Y}_i^\top \parallel \mathbf{x}_i) \mathbf{A} \in \{0, 1\}^{\lambda \times \lambda}$

$\mathbf{D}_i = \mathbf{Y}_i^\top + \mathbf{x}_i \cdot \tilde{\mathbf{r}}^\top \in \{0, 1\}^{\lambda \times (\lambda-1)}$ ,  $\mathbf{Z}_i = (\mathbf{0} \parallel \mathbf{D}_i) \mathbf{R}_0^\top \in \{0, 1\}^{\lambda \times \lambda}$

$\mathbf{y}' \xleftarrow{\$} \{0, 1\}^{\lambda-1}$ ,  $\mathbf{z}' = (\mathbf{y}'^\top \parallel x') \mathbf{A} \in \{0, 1\}^{1 \times \lambda}$

$\mathbf{d}' = \mathbf{y}'^\top + x' \cdot \tilde{\mathbf{r}}^\top \in \{0, 1\}^{1 \times (\lambda-1)}$ ,  $\mathbf{z}' = (\mathbf{0} \parallel \mathbf{d}') \mathbf{R}_0^\top \in \{0, 1\}^{1 \times \lambda}$

$\text{pk} = (\mathbf{A}, (\mathbf{Z}_i)_{1 \leq i \leq \ell}, \mathbf{z}')$ ,  $\text{sk} = (\text{sk}_{\text{MAC}}, (\mathbf{Y}_i)_{1 \leq i \leq \ell}, \mathbf{y}')$

Return pk

FINALIZE( $\beta$ ): //Games  $G_0$ - $G_6$

If  $(\text{pk}(x, y) \neq 1$  for all  $y \in \mathcal{Q}_y$ , return  $\beta$

Else return 0

USKGEN( $y$ ): //Games  $G_0$ - $G_4$ ,  $G_5$ - $G_6$

$\mathcal{Q}_y = \mathcal{Q}_y \cup \{y\}$ ,  $(\mathbf{t}, \mathbf{u}) \xleftarrow{\$} \text{Tag}_\lambda(\text{sk}_{\text{MAC}}, y)$

$\mathbf{v} = \text{rE}_\lambda(y, \begin{pmatrix} \mathbf{t}^\top \mathbf{Y}_1^\top \\ \vdots \\ \mathbf{t}^\top \mathbf{Y}_\ell^\top \end{pmatrix}) + \text{kE}_\lambda(y, \mathbf{y}'^\top) \in \{0, 1\}^{\eta \times (\lambda-1)}$

$\mathbf{v} = \text{rE}_\lambda(y, (\mathbf{D}_1^\top \mathbf{t}, \dots, \mathbf{D}_\ell^\top \mathbf{t})^\top) + \text{kE}_\lambda(y, \mathbf{d}') - \mathbf{u} \cdot \tilde{\mathbf{r}}^\top \in \{0, 1\}^{\eta \times (\lambda-1)}$

$\text{usk}[y] = (\mathbf{t}, \mathbf{u}, \mathbf{v})$

Return  $\text{usk}[y]$

ENC( $x$ ): //Games  $G_0$ ,  $G_1$ - $G_4$ ,  $G_3$ - $G_4$ ,  $G_5$ ,  $G_6$

$\mathbf{r} \xleftarrow{\$} \{0\} \times \{0, 1\}^{\lambda-1}$ ,  $\mathbf{r} \xleftarrow{\$} \{1\} \times \{0, 1\}^{\lambda-1}$

$\mathbf{c}_0^* = \mathbf{A} \mathbf{r} \in \{0, 1\}^\lambda$ ,  $\mathbf{c}_0^* = (\mathbf{A} + \mathbf{N}^\lambda) \mathbf{r}$

$\mathbf{C}_1^* = \text{sE}_\lambda(x, \begin{pmatrix} \mathbf{r}^\top \mathbf{Z}_1^\top \\ \vdots \\ \mathbf{r}^\top \mathbf{Z}_\ell^\top \end{pmatrix}) \in \{0, 1\}^{\zeta \cdot \lambda}$

$\mathbf{C}_1^* = \text{sE}_\lambda(x, ((\mathbf{Y}_1^\top \parallel \mathbf{x}_1)(\mathbf{A} + \mathbf{N}^\lambda) \mathbf{r}, \dots, (\mathbf{Y}_\ell^\top \parallel \mathbf{x}_\ell)(\mathbf{A} + \mathbf{N}^\lambda) \mathbf{r})^\top)$

$\mathbf{C}_1^* = \text{sE}_\lambda(x, (\mathbf{Z}_1 \mathbf{r}, \dots, \mathbf{Z}_\ell \mathbf{r})^\top) + \text{sE}_\lambda(x, (\mathbf{x}_1, \dots, \mathbf{x}_\ell)^\top)$

$\mathbf{K}^* = \mathbf{z}' \cdot \mathbf{r} \in \{0, 1\}$ ,  $\mathbf{K}^* = (\mathbf{y}'^\top \parallel x')(\mathbf{A} + \mathbf{N}^\lambda) \mathbf{r}$ ,  $\mathbf{K}^* = \mathbf{z}' \cdot \mathbf{r} + x'$

$\mathbf{K}^* \xleftarrow{\$} \{0, 1\}$

Return  $\mathbf{K}^*$  and  $\text{ct}^* = (\mathbf{c}_0^*, \mathbf{C}_1^*)$

Fig. 13. Games  $G_0$ - $G_6$  for the proof of Theorem 4.



**Lemma 13.** *There exists an adversary  $\mathcal{B}_1 = \{b_\lambda^1\}_{\lambda \in \mathbb{N}} \in \text{NC}^1$  such that  $b_\lambda^1$  breaks the fine-grained matrix linear assumption (see Definition 5), which holds under  $\text{NC}^1 \subsetneq \oplus \text{L}/\text{poly}$  according to Theorem 1, with advantage*

$$|\Pr[\mathbf{G}_2^{a_\lambda} \Rightarrow 1] - \Pr[\mathbf{G}_1^{a_\lambda} \Rightarrow 1]|.$$

*Proof.*  $\mathbf{G}_1$  and  $\mathbf{G}_2$  only differ in the distribution of  $\mathbf{A}$ , namely,  $\mathbf{A}^\top \stackrel{\$}{\leftarrow} \text{ZeroSamp}(\lambda)$  or  $\mathbf{A}^\top \stackrel{\$}{\leftarrow} \text{OneSamp}(\lambda)$ , and we build the distinguisher  $b_\lambda^1$  as follows.

$b_\lambda^1$  runs in exactly the same way as the challenger of  $\mathbf{G}_1$  except that in INIT, instead of generating  $\mathbf{A}$  by itself, it takes as input  $\mathbf{A}^\top$  generated as  $\mathbf{A}^\top \stackrel{\$}{\leftarrow} \text{ZeroSamp}(\lambda)$  or  $\mathbf{A}^\top \stackrel{\$}{\leftarrow} \text{OneSamp}(\lambda)$  from its own challenger. When  $a_\lambda$  outputs  $\beta$ ,  $b_\lambda^1$  outputs  $\beta$  as well if no  $y$  such that  $\rho_\lambda(x, y) = 1$  was queried to  $\text{USKGEN}$ . Otherwise,  $b_\lambda^1$  outputs 0.

If  $\mathbf{A}$  is generated as  $\mathbf{A}^\top \stackrel{\$}{\leftarrow} \text{ZeroSamp}(\lambda)$  (respectively,  $\mathbf{A}^\top \stackrel{\$}{\leftarrow} \text{OneSamp}(\lambda)$ ), the view of  $a_\lambda$  is the same as its view in  $\mathbf{G}_1$  (respectively,  $\mathbf{G}_2$ ). Hence, the probability that  $b_\lambda^1$  breaks the fine-grained matrix linear assumption is

$$|\Pr[\mathbf{G}_2^{a_\lambda} \Rightarrow 1] - \Pr[\mathbf{G}_1^{a_\lambda} \Rightarrow 1]|.$$

Moreover, since  $a_\lambda$  only makes constant rounds of queries, all operations in  $b_\lambda^1$  are performed in  $\text{NC}^1$ . Hence, we have  $\mathcal{B}_1 = \{b_\lambda^1\}_{\lambda \in \mathbb{N}} \in \text{NC}^1$ , completing this part of proof.  $\square$

**Lemma 14.**  $\Pr[\mathbf{G}_3^{a_\lambda} \Rightarrow 1] = \Pr[\mathbf{G}_2^{a_\lambda} \Rightarrow 1]$ .

*Proof.* In this game, we sample  $\mathbf{r}$  in  $\text{ENC}(x)$  as  $\mathbf{r} \stackrel{\$}{\leftarrow} \{0, 1\}^\lambda$  instead of  $\mathbf{r} \stackrel{\$}{\leftarrow} \{0\} \times \{0, 1\}^{\lambda-1}$ . According to Lemma 4, the distributions of  $\mathbf{A} + \mathbf{N}^\lambda$  in both  $\mathbf{G}_2$  and  $\mathbf{G}_3$  are identical to that of a matrix sampled from  $\text{ZeroSamp}$ . Then this lemma follows from Lemma 6 immediately.  $\square$

**Lemma 15.** *There exists an adversary  $\mathcal{B}_2 = \{b_\lambda^2\}_{\lambda \in \mathbb{N}} \in \text{NC}^1$  such that  $b_\lambda^2$  breaks the fine-grained matrix linear assumption with advantage*

$$|\Pr[\mathbf{G}_4^{a_\lambda} \Rightarrow 1] - \Pr[\mathbf{G}_3^{a_\lambda} \Rightarrow 1]|.$$

*Proof.*  $\mathbf{G}_1$  and  $\mathbf{G}_2$  only differ in the distribution of  $\mathbf{A}$ , namely,  $\mathbf{A}^\top \stackrel{\$}{\leftarrow} \text{OneSamp}(\lambda)$  or  $\mathbf{A}^\top \stackrel{\$}{\leftarrow} \text{ZeroSamp}(\lambda)$ , and we build the distinguisher  $b_\lambda^2$  against Lemma 1 as follows.

$b_\lambda^2$  runs in exactly the same way as the challenger of  $\mathbf{G}_3$  except that in INIT, instead of generating  $\mathbf{A}$  by itself, it takes as input  $\mathbf{A}^\top$  generated as  $\mathbf{A}^\top \stackrel{\$}{\leftarrow} \text{ZeroSamp}(\lambda)$  or  $\mathbf{A}^\top \stackrel{\$}{\leftarrow} \text{OneSamp}(\lambda)$  from its own challenger. When  $a_\lambda$  outputs  $\beta$ ,  $b_\lambda^2$  outputs  $\beta$  as well if no  $y$  such that  $\rho_\lambda(x, y) = 1$  was queried to  $\text{USKGEN}$ . Otherwise,  $b_\lambda^2$  outputs 0.

If  $\mathbf{A}$  is generated as  $\mathbf{A}^\top \stackrel{\$}{\leftarrow} \text{OneSamp}(\lambda)$  (respectively,  $\mathbf{A}^\top \stackrel{\$}{\leftarrow} \text{ZeroSamp}(\lambda)$ ), the view of  $a_\lambda$  is the same as its view in  $\mathbf{G}_3$  (respectively,  $\mathbf{G}_4$ ). Hence, the probability that  $b_\lambda^2$  breaks the fine-grained matrix linear assumption is

$$|\Pr[\mathbf{G}_4^{a_\lambda} \Rightarrow 1] - \Pr[\mathbf{G}_3^{a_\lambda} \Rightarrow 1]|.$$

Moreover, since  $a_\lambda$  only makes constant rounds of queries, all operations in  $b_\lambda^2$  are performed in  $\text{NC}^1$ . Hence, we have  $\mathcal{B}_2 = \{b_\lambda^2\}_{\lambda \in \mathbb{N}} \in \text{NC}^1$ , completing this part of proof.  $\square$

**Lemma 16.**  $\Pr[\mathbf{G}_5^{a_\lambda} \Rightarrow 1] = \Pr[\mathbf{G}_4^{a_\lambda} \Rightarrow 1]$ .

*Proof.* In  $\mathbf{G}_5$ , we do not use  $(\mathbf{Y}_i)_{i=1}^\ell$  and  $\mathbf{y}'$  in  $\text{USKGEN}(\mathbf{y})$  or  $\text{ENC}(\mathbf{x})$  any more. We give the sampling procedure for  $\mathbf{A}$  in an explicit way and change the simulation of  $\mathbf{Z}_i$ ,  $\mathbf{z}'$ ,  $\mathbf{v}$ ,  $\mathbf{C}_1^*$ , and  $\mathbf{K}^*$  as in Figure 13. We now show that all the changes are purely conceptual.

In  $\mathbf{G}_5$ , we generate  $\mathbf{A}$  by sampling  $\mathbf{R}_1 = \begin{pmatrix} \mathbf{I}_{\lambda-1} & 0 \\ \tilde{\mathbf{r}}^\top & 1 \end{pmatrix}^\top \stackrel{\$}{\leftarrow} \text{RSamp}(\lambda)$  and  $\mathbf{R}_0 \stackrel{\$}{\leftarrow} \text{LSamp}(\lambda)$ , and setting  $\mathbf{A}^\top = \mathbf{R}_0 \mathbf{M}_0^\lambda \mathbf{R}_1$ . This is exactly the “zero-sampling” procedure, in which case, we have

$$\begin{aligned} \mathbf{Z}_i &= (\mathbf{Y}_i^\top \parallel \mathbf{x}_i) \mathbf{A} = (\mathbf{Y}_i^\top \parallel \mathbf{x}_i) \mathbf{R}_1^\top \mathbf{M}_0^\lambda \mathbf{R}_0^\top \\ &= (\mathbf{Y}_i^\top \parallel \mathbf{x}_i) \begin{pmatrix} \mathbf{I}_{\lambda-1} & \mathbf{0} \\ \tilde{\mathbf{r}}^\top & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & & 1 \\ 0 & & \cdots & & 0 \end{pmatrix} \mathbf{R}_0^\top \\ &= (\mathbf{Y}_i^\top + \mathbf{x}_i \cdot \tilde{\mathbf{r}}^\top \parallel \mathbf{x}_i) \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & & 1 \\ 0 & & \cdots & & 0 \end{pmatrix} \mathbf{R}_0^\top \\ &= (\mathbf{0} \parallel \mathbf{Y}_i^\top + \mathbf{x}_i \cdot \tilde{\mathbf{r}}^\top) \mathbf{R}_0^\top = (\mathbf{0} \parallel \mathbf{D}_i) \mathbf{R}_0^\top \end{aligned}$$

and

$$\begin{aligned} \mathbf{C}_1^* &= \text{sE}_\lambda(\mathbf{x}, ((\mathbf{Y}_1^\top \parallel \mathbf{x}_1)(\mathbf{A} + \mathbf{N}^\lambda) \mathbf{r}, \dots, (\mathbf{Y}_\ell^\top \parallel \mathbf{x}_\ell)(\mathbf{A} + \mathbf{N}^\lambda) \mathbf{r})^\top) \\ &= \text{sE}_\lambda(\mathbf{x}, (\mathbf{Z}_1 \mathbf{r} + \mathbf{x}_1, \dots, \mathbf{Z}_\ell \mathbf{r} + \mathbf{x}_\ell)^\top) \\ &= \text{sE}_\lambda(\mathbf{x}, (\mathbf{Z}_1 \mathbf{r}, \dots, \mathbf{Z}_\ell \mathbf{r})^\top) + \text{sE}_\lambda(\mathbf{x}, (\mathbf{x}_1, \dots, \mathbf{x}_\ell)^\top). \end{aligned}$$

Hence, the distributions of  $\mathbf{Z}_i$  in  $\mathbf{G}_5$  remain the same, and the distributions of  $\mathbf{z}'$  and  $\mathbf{K}^*$  can be analyzed in the same way. The distribution of  $\mathbf{v}$  does not change as well since

$$\begin{aligned} \mathbf{v} &= \text{rE}_\lambda(\mathbf{y}, (\mathbf{Y}_1 \mathbf{t}, \dots, \mathbf{Y}_\ell \mathbf{t})^\top) + \text{kE}_\lambda(\mathbf{y}, \mathbf{y}'^\top) \\ &= \text{rE}_\lambda(\mathbf{y}, ((\mathbf{Y}_1 + \tilde{\mathbf{r}} \cdot \mathbf{x}_1^\top) \mathbf{t}, \dots, (\mathbf{Y}_\ell + \tilde{\mathbf{r}} \cdot \mathbf{x}_\ell^\top) \mathbf{t})^\top) + \text{kE}_\lambda(\mathbf{y}, \mathbf{y}'^\top + \mathbf{x}' \cdot \tilde{\mathbf{r}}^\top) \\ &\quad - (\text{rE}_\lambda(\mathbf{y}, (\tilde{\mathbf{r}} \cdot \mathbf{x}_1^\top \cdot \mathbf{t}, \dots, \tilde{\mathbf{r}} \cdot \mathbf{x}_\ell^\top \cdot \mathbf{t})^\top) + \text{kE}_\lambda(\mathbf{y}, \mathbf{x}' \cdot \tilde{\mathbf{r}}^\top)) \\ &= \text{rE}_\lambda(\mathbf{y}, (\mathbf{D}_1^\top \mathbf{t}, \dots, \mathbf{D}_\ell^\top \mathbf{t})^\top) + \text{kE}_\lambda(\mathbf{y}, \mathbf{d}') - \mathbf{u} \cdot \tilde{\mathbf{r}}^\top. \end{aligned}$$

Putting all above together, Lemma 16 immediately follows.  $\square$

**Lemma 17.** *There exists an adversary  $\mathcal{B}_3 = \{b_\lambda^3\}_{\lambda \in \mathbb{N}} \in \text{NC}^1$  such that  $b_\lambda^3$  breaks the  $\text{NC}^1$ - $(k, l)$ -PR-CMA security of  $\text{MAC}_{\text{GA}}$  with advantage*

$$|\Pr[\mathbf{G}_6^{a_\lambda} \Rightarrow 1] - \Pr[\mathbf{G}_5^{a_\lambda} \Rightarrow 1]|.$$

*Proof.* The challenger of  $\mathbf{G}_6$  answers the  $\text{ENC}(x)$  query by choosing random  $\mathbf{K}^*$ . We build  $b_\lambda^3$  as in Figure 14 to show that the differences between  $\mathbf{G}_6$  and  $\mathbf{G}_5$  can be bounded by its advantage of breaking the PR-CMA security of  $\text{MAC}_{\text{GA}}$ .

$b_\lambda^3$  runs in the same way as the challenger of  $\mathbf{G}_5$  except that it samples  $\mathbf{D}_i$  and  $\mathbf{d}'$  uniformly at random from  $\{0, 1\}^{\lambda \times (\lambda-1)}$  and  $\{0, 1\}^{1 \times (\lambda-1)}$  respectively. This does not change the view of  $a_\lambda$  since  $\mathbf{Y}_i$  and  $\mathbf{y}'$  were uniformly sampled in  $\mathbf{G}_5$ . Moreover, every time on receiving a query  $\mathbf{y}$  to  $\text{USKGEN}$ ,  $b_\lambda^3$  forwards  $\mathbf{y}$  to its evaluation oracle  $\text{EVAL}$  to obtain the answer  $(\mathbf{t}, \mathbf{u})$ , and on receiving the query  $x$  to  $\text{ENC}$ ,  $b_\lambda^3$  forwards  $x$  to its challenge oracle  $\text{CHAL}$  and uses the answer  $(h, \mathbf{h}_0, h_1)$  to simulate  $\mathbf{r}$ ,  $\mathbf{C}_1^*$ , and  $\mathbf{K}^*$  as in Figure 14. When  $a_\lambda$  outputs  $\beta$ ,  $b_\lambda^3$  outputs  $\beta$  as well if no  $\mathbf{y}$  such that  $\rho_\lambda(x, \mathbf{y}) = 1$  was queried to  $\text{USKGEN}$ . Otherwise,  $b_\lambda^3$  outputs 0.

<p><u>INIT:</u></p> $\mathbf{R}_1 = \begin{pmatrix} \mathbf{I}_{\lambda-1} & \mathbf{0} \\ \tilde{\mathbf{r}}^\top & 1 \end{pmatrix} \stackrel{\$}{\leftarrow} \text{RSamp}(\lambda),$ $\mathbf{R}_0 \stackrel{\$}{\leftarrow} \text{LSamp}(\lambda), \mathbf{A}^\top = \mathbf{R}_0 \mathbf{M}_0^\lambda \mathbf{R}_1$ <p>For <math>i = 1, \dots, \ell</math>:</p> $\mathbf{D}_i \stackrel{\$}{\leftarrow} \{0, 1\}^{\lambda \times (\lambda-1)}$ $\mathbf{Z}_i = (\mathbf{0} \parallel \mathbf{D}_i) \mathbf{R}_0^\top \in \{0, 1\}^{\lambda \times \lambda}$ $\mathbf{d}' \stackrel{\$}{\leftarrow} \{0, 1\}^{1 \times (\lambda-1)}, \mathbf{z}' = (\mathbf{0} \parallel \mathbf{d}') \mathbf{R}_0^\top \in \{0, 1\}^{1 \times \lambda}$ $\text{pk} = (\mathbf{A}, (\mathbf{Z}_i)_{1 \leq i \leq \ell}, \mathbf{z}')$ <p>Return pk</p> <p><u>USKGEN(y):</u></p> $\mathcal{Q}_y = \mathcal{Q}_y \cup \{\mathbf{y}\}$ $(\mathbf{t}, \mathbf{u}) \stackrel{\$}{\leftarrow} \text{EVAL}(\mathbf{y})$ $\mathbf{v} = \mathbf{rE}_\lambda(\mathbf{y}, (\mathbf{D}_1^\top \mathbf{t}, \dots, \mathbf{D}_\ell^\top \mathbf{t})^\top) + \mathbf{kE}_\lambda(\mathbf{y}, \mathbf{d}') - \mathbf{u}$ $\tilde{\mathbf{r}}^\top \in \{0, 1\}^{\eta \times (\lambda-1)}$ $\text{usk}[\mathbf{y}] = (\mathbf{t}, \mathbf{u}, \mathbf{v})$ <p>Return usk[y]</p>	<p><u>ENC(x):</u> //one query</p> $(\mathbf{h}_0, h_1) \stackrel{\$}{\leftarrow} \text{CHAL}(x)$ $r_2, \dots, r_n \stackrel{\$}{\leftarrow} \{0, 1\}$ $\mathbf{r} = (1, r_2, \dots, r_n)^\top$ $\mathbf{c}_0^* = (\mathbf{A} + \mathbf{N}^\lambda) \mathbf{r} \in \{0, 1\}^\lambda$ $\mathbf{C}_1^* = \mathbf{sE}_\lambda(x, \begin{pmatrix} \mathbf{r}^\top \mathbf{Z}_1^\top \\ \vdots \\ \mathbf{r}^\top \mathbf{Z}_\ell^\top \end{pmatrix}) + \mathbf{h}_0 \in$ $\{0, 1\}^{\zeta \times \lambda}$ $\mathbf{K}^* = \mathbf{z}' \cdot \mathbf{r} + h_1 \in \{0, 1\}$ <p>Return <math>\mathbf{K}^*</math> and <math>\text{ct}^* = (\mathbf{c}_0^*, \mathbf{C}_1^*)</math></p> <p><u>FINALIZE(<math>\beta</math>):</u></p> <p>If <math>(\rho_\lambda(x, \mathbf{y}) \neq 1</math> for all <math>\mathbf{y} \in \mathcal{Q}_y</math>)</p> <p style="padding-left: 20px;">return <math>\beta</math></p> <p>Else return 0</p>
--	---

**Fig. 14.** Description of  $\mathcal{B}_3 = \{b_\lambda^3\}_{\lambda \in \mathbb{N}}$  (having access to the oracles  $\text{INIT}_{\text{MAC}}, \text{EVAL}, \text{CHAL}, \text{FINALIZE}_{\text{MAC}}$  of the  $\text{PR-CMA}_{\text{real}}/\text{PR-CMA}_{\text{rand}}$  games of Figure 5) for the proof of Lemma 17.

If  $h_1$  is uniform (i.e.,  $b_\lambda^3$  is in Game  $\text{PR-CMA}_{\text{rand}}$ ) then the view of  $a_\lambda$  is identical to its view in  $\mathbf{G}_6$ . If  $h_1$  is real (i.e.,  $b_\lambda^3$  is in Game  $\text{PR-CMA}_{\text{real}}$ ) then the view of  $\mathcal{A}$  is identical to its view in  $\mathbf{G}_5$ . Hence, the advantage of  $b_\lambda^3$  in breaking

the PR-CMA security is

$$|\Pr[G_6^{a_\lambda} \Rightarrow 1] - \Pr[G_5^{a_\lambda} \Rightarrow 1]|.$$

Moreover, since  $a_\lambda$  only makes constant rounds of queries, all operations in  $b_\lambda^3$  are performed in  $\text{NC}^1$ . Hence, we have  $\mathcal{B}_3 = \{b_\lambda^3\}_{\lambda \in \mathbb{N}} \in \text{NC}^1$ , completing this part of proof.

We now do all the previous steps in the reverse order as in Figure 15. Note that the view of the adversary in  $H_0$  (respectively,  $H_4$ ) is identical to its view in  $G_6$  (respectively,  $\text{PR-AT-CPA}_{\text{rand}}$ ). By using the above arguments in a reverse order, we have the following lemma.

**Lemma 18.** *There exists an adversary  $\mathcal{B}_4 = \{b_\lambda^4\}_{\lambda \in \mathbb{N}} \in \text{NC}^1$  such that  $b_\lambda^4$  breaks the fine-grained matrix linear assumption with advantage*

$$(|\Pr[H_4^{a_\lambda} \Rightarrow 1] - \Pr[H_0^{a_\lambda} \Rightarrow 1]|)/2.$$

□

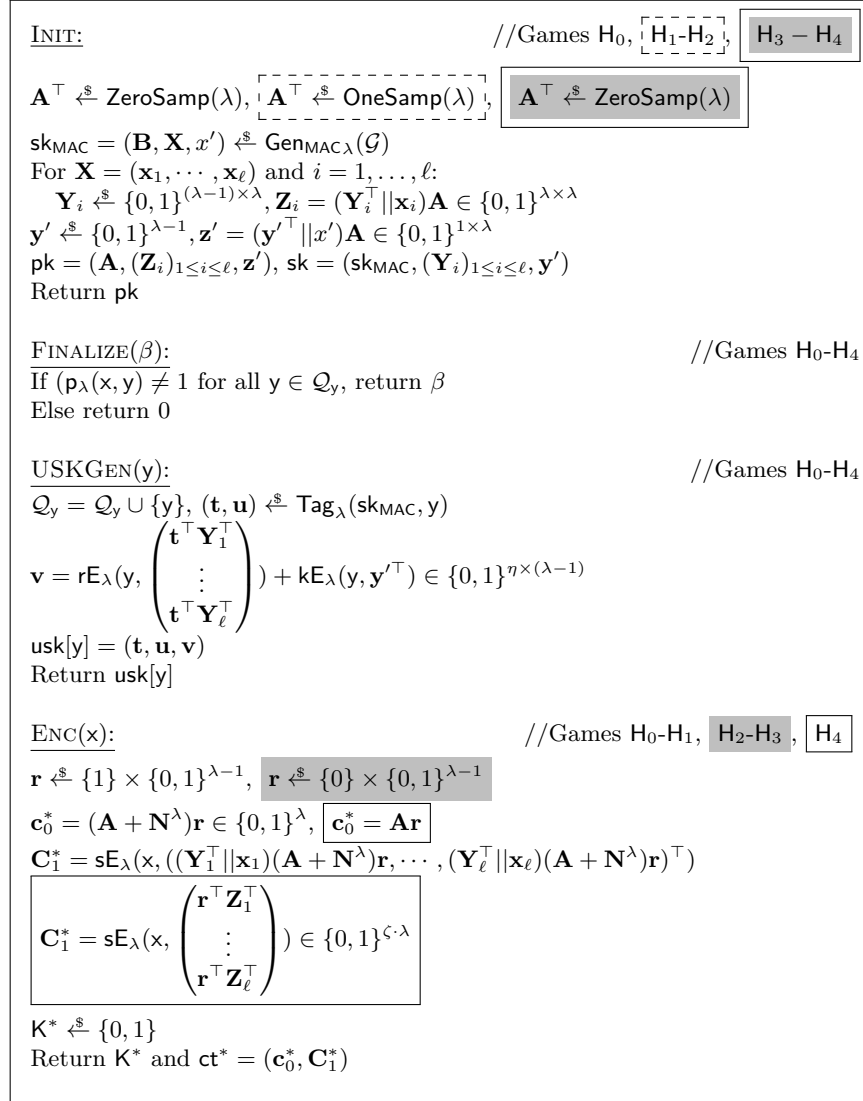
Putting all above together, Theorem 4 immediately follows. □

## Acknowledgements

We would like to thank the anonymous reviewers for their valuable comments on a previous version of this paper. Yuyu Wang was supported by the National Natural Science Foundation for Young Scientists of China under Grant Number 62002049, the Fundamental Research Funds for the Central Universities under Grant Number ZYGX2020J017, and the Sichuan Science and Technology Program under Grant Numbers 2019YFG0506 and 2020YFG0292. Yu Chen was supported by the National Natural Science Foundation of China under Grant Numbers 61772522 and 61932019.

## References

1. Applebaum, B., Ishai, Y., Kushilevitz, E.: Cryptography in  $\text{NC}^0$ . In: 45th FOCS. pp. 166–175. IEEE Computer Society Press (Oct 2004) 3, 4
2. Ball, M., Dachman-Soled, D., Kulkarni, M.: New techniques for zero-knowledge: Leveraging inefficient provers to reduce assumptions, interaction, and trust. In: Micciancio, D., Ristenpart, T. (eds.) CRYPTO 2020, Part III. LNCS, vol. 12172, pp. 674–703. Springer, Heidelberg (Aug 2020) 2
3. Barrington, D.A.M.: Bounded-width polynomial-size branching programs recognize exactly those languages in  $\text{NC}^1$ . In: 18th ACM STOC. pp. 1–5. ACM Press (May 1986) 2
4. Bellare, M., Goldwasser, S.: New paradigms for digital signatures and message authentication based on non-interactive zero knowledge proofs. In: Brassard, G. (ed.) CRYPTO’89. LNCS, vol. 435, pp. 194–211. Springer, Heidelberg (Aug 1990) 4
5. Blazy, O., Kiltz, E., Pan, J.: (Hierarchical) identity-based encryption from affine message authentication. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014, Part I. LNCS, vol. 8616, pp. 408–425. Springer, Heidelberg (Aug 2014) 2, 3, 5, 6, 11, 13

Fig. 15. Games  $H_0$ - $H_4$  for the proof of Theorem 4.

6. Boneh, D., Franklin, M.K.: Identity-based encryption from the Weil pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (Aug 2001) 2
7. Boneh, D., Papakonstantinou, P.A., Rackoff, C., Vahlis, Y., Waters, B.: On the impossibility of basing identity based encryption on trapdoor permutations. In: 49th FOCS. pp. 283–292. IEEE Computer Society Press (Oct 2008) 2
8. Brzuska, C., Couteau, G.: Towards fine-grained one-way functions from strong average-case hardness. IACR Cryptol. ePrint Arch. 2020, 1326 (2020) 2
9. Campanelli, M., Gennaro, R.: Fine-grained secure computation. In: Beimel, A., Dziembowski, S. (eds.) TCC 2018, Part II. LNCS, vol. 11240, pp. 66–97. Springer, Heidelberg (Nov 2018) 2, 3
10. Chen, J., Gay, R., Wee, H.: Improved dual system ABE in prime-order groups via predicate encodings. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015, Part II. LNCS, vol. 9057, pp. 595–624. Springer, Heidelberg (Apr 2015) 2, 3, 4, 5, 9, 10, 20
11. Chen, J., Wee, H.: Fully, (almost) tightly secure IBE and dual system groups. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part II. LNCS, vol. 8043, pp. 435–460. Springer, Heidelberg (Aug 2013) 3
12. Cocks, C.: An identity based encryption scheme based on quadratic residues. In: Honary, B. (ed.) 8th IMA International Conference on Cryptography and Coding. LNCS, vol. 2260, pp. 360–363. Springer, Heidelberg (Dec 2001) 2
13. Degwekar, A., Vaikuntanathan, V., Vasudevan, P.N.: Fine-grained cryptography. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016, Part III. LNCS, vol. 9816, pp. 533–562. Springer, Heidelberg (Aug 2016) 2, 3, 7
14. Egashira, S., Wang, Y., Tanaka, K.: Fine-grained cryptography revisited. In: Galbraith, S.D., Moriai, S. (eds.) ASIACRYPT 2019, Part III. LNCS, vol. 11923, pp. 637–666. Springer, Heidelberg (Dec 2019) 2, 3, 5
15. Egashira, S., Wang, Y., Tanaka, K.: Fine-grained cryptography revisited. J. Cryptol. 34(3), 23 (2021) 4, 8, 9
16. Escala, A., Herold, G., Kiltz, E., Ràfols, C., Villar, J.: An algebraic framework for Diffie-Hellman assumptions. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part II. LNCS, vol. 8043, pp. 129–147. Springer, Heidelberg (Aug 2013) 3
17. Fuchsbaauer, G., Kiltz, E., Loss, J.: The algebraic group model and its applications. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018, Part II. LNCS, vol. 10992, pp. 33–62. Springer, Heidelberg (Aug 2018) 2
18. Gentry, C., Silverberg, A.: Hierarchical ID-based cryptography. In: Zheng, Y. (ed.) ASIACRYPT 2002. LNCS, vol. 2501, pp. 548–566. Springer, Heidelberg (Dec 2002) 5
19. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: Juels, A., Wright, R.N., De Capitani di Vimercati, S. (eds.) ACM CCS 2006. pp. 89–98. ACM Press (Oct / Nov 2006), available as Cryptology ePrint Archive Report 2006/309 2
20. Groth, J., Sahai, A.: Efficient non-interactive proof systems for bilinear groups. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 415–432. Springer, Heidelberg (Apr 2008) 4
21. Horwitz, J., Lynn, B.: Toward hierarchical identity-based encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 466–481. Springer, Heidelberg (Apr / May 2002) 5
22. Ishai, Y., Kushilevitz, E.: Randomizing polynomials: A new representation with applications to round-efficient secure computation. In: 41st FOCS. pp. 294–304. IEEE Computer Society Press (Nov 2000) 3, 4

23. Jutla, C.S., Roy, A.: Shorter quasi-adaptive NIZK proofs for linear subspaces. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013, Part I. LNCS, vol. 8269, pp. 1–20. Springer, Heidelberg (Dec 2013) 3
24. Lewko, A.B., Waters, B.: New techniques for dual system encryption and fully secure HIBE with short ciphertexts. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 455–479. Springer, Heidelberg (Feb 2010) 5
25. Maurer, U.M.: Abstract models of computation in cryptography (invited paper). In: Smart, N.P. (ed.) 10th IMA International Conference on Cryptography and Coding. LNCS, vol. 3796, pp. 1–12. Springer, Heidelberg (Dec 2005) 2
26. Merkle, R.C.: Secure communications over insecure channels. *Commun. ACM* 21(4), 294–299 (1978) 2
27. Razborov, A.A.: Lower bounds on the size of bounded depth circuits over a complete basis with logical addition. *Mathematical notes of the Academy of Sciences of the USSR* 41(4) (Apr 1987) 3, 7
28. Shamir, A.: Identity-based cryptosystems and signature schemes. In: Blakley, G.R., Chaum, D. (eds.) CRYPTO’84. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (Aug 1984) 2
29. Shoup, V.: Lower bounds for discrete logarithms and related problems. In: Fumy, W. (ed.) EUROCRYPT’97. LNCS, vol. 1233, pp. 256–266. Springer, Heidelberg (May 1997) 2
30. Smolensky, R.: Algebraic methods in the theory of lower bounds for Boolean circuit complexity. In: Aho, A. (ed.) 19th ACM STOC. pp. 77–82. ACM Press (May 1987) 3, 7
31. Waters, B.: Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 619–636. Springer, Heidelberg (Aug 2009) 5
32. Wee, H.: Dual system encryption via predicate encodings. In: Lindell, Y. (ed.) TCC 2014. LNCS, vol. 8349, pp. 616–637. Springer, Heidelberg (Feb 2014) 3, 4, 5, 20