Fine-Grained Secure Attribute-Based Encryption*

Yuyu Wang [1] Jiaxin Pan [2] Yu Chen [3,4,5]

September 6, 2023

[1] University of Electronic Science and Technology of China, Chengdu, China
wangyuyu@uestc.edu.cn
[2] Department of Mathematical Sciences,
NTNU - Norwegian University of Science and Technology, Trondheim, Norway
jiaxin.pan@ntnu.no
[3] School of Cyber Science and Technology, Shandong University, Qingdao 266237, China
[4] State Key Laboratory of Cryptology, P.O. Box 5159, Beijing 100878, China
[5] Key Laboratory of Cryptologic Technology and Information Security, Ministry of Education,
Shandong University, Qingdao 266237, China
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, $\mathsf{NC}^1 \subsetneq \oplus\mathsf{L}/\mathsf{poly}$, 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 [30, 25] or algebraic groups [16]). 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

*A preliminary version of this paper appeared at Crypto 2021 [34], this is the full version.

assumption extremely mild, for instance, assuming $\mathsf{NC}^1 \subsetneq \oplus\mathsf{L}/\mathsf{poly}$. This is a widely accepted worst-case assumption. As $\oplus\mathsf{L}/\mathsf{poly}$ is the class of languages with polynomial-sized branching programs and all languages in 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 from 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, $\mathsf{NC}^1 \subsetneq \oplus\mathsf{L}/\mathsf{poly}$ 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 (leveled fully homomorphic) public-key encryption [13, 9], verifiable computation [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 [18] 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 $\mathsf{p}(\mathsf{x}, \mathsf{y}) = 1$ 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 [29, 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 develop new techniques to improve on the state of the art of fine-grained cryptography, which only provides primitives related to TDP and CCA-secure PKE.

1.2 Our Contributions

We construct the *first* fine-grained secure ABE scheme. In particular, our scheme is computable in $\mathsf{AC}^0[2]$ and secure against adversaries in NC^1. Note that $\mathsf{AC}^0[2] \subsetneq \mathsf{NC}^1$ [28, 31]. 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 $\mathsf{NC}^1 \subsetneq \oplus\mathsf{L}/\mathsf{poly}$. 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 $\mathsf{NC}^1 \subsetneq \oplus\mathsf{L}/\mathsf{poly}$ 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 [36, 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 [22] with fine-grained security. Here "quasi-adaptive" means that common reference strings (CRSs) may depend on the language of the NIZK system.

Applications of security against NC^1. Other than only relying on weak assumptions and running with low complexity, our results have the following applications.

Since security against NC^1 captures adversaries with limited parallel running-time, our constructions are well-suited for systems where attacks make sense only if they succeed in a short period of time. For example, our ABEs (and other fine-grained encryption primitives) can be used to protect messages that are only valuable in a short period of time, and that can be published or deleted later. As another

example, the fine-grained signature (implied by our fine-grained IBE) and fine-grained QA-NIZK prevent adversaries from forging signatures and proofs with the running-time of an honest user, thereby ensuring security by letting the system reject users who have timed out when attempting to generate signatures or proofs. Moreover, as noted in [13], combining fine-grained primitives with standard ones immediately yield hybrids that are secure against NC^1 adversaries under $\mathsf{NC}^1 \subsetneq \oplus\mathsf{L}/\mathsf{poly}$ and secure against polynomial time adversaries under stronger assumptions.

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 [21, 1, 14] in achieving our goal. In a nutshell, we transform a suitable symmetric-key primitive to an ABE in the fine-grained setting.

Previous frameworks in [5, 10] use pairings and the Diffie-Hellman assumptions. In contrast to them, our work develops new techniques to build ABEs without pairings or the Diffie-Hellman assumptions, but only under the mild assumption that $\mathsf{NC}^1 \subsetneq \oplus\mathsf{L}/\mathsf{poly}$. 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 Blazy, Kiltz, and Pan, and its limitations in NC^1. The "MAC→IBE" transformation of BKP [5] is an abstraction of the Chen-Wee (CW) IBE scheme [11], and it also implements 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, $\mathsf{pk} = \mathsf{Com}(\mathsf{sk}_{\mathsf{MAC}})$, is a commitment of the MAC secret key, $\mathsf{sk}_{\mathsf{MAC}}$. A user secret key $\mathsf{usk}[\mathsf{id}]$ of an identity id consists of a BG signature, namely, a MAC tag τ_{id} on the message id and a NIZK proof of the validity of τ_{id} w.r.t. the secret key committed in pk.

Since the MAC verification consists of only affine equations, after implementing the aforementioned commitments and NIZK proofs with a (tuned) Groth-Sahai (GS) proof system [19][1], the BKP IBE ciphertext $\mathsf{ct}_{\mathsf{id}}$ can be viewed as a randomized linear combination of pk w.r.t. id. This is the key observation of BKP. The BKP framework can be further improved and extended to construct ABEs using predicate encodings [36] as in the CGW framework [10] by Chen, Gay, and Wee.

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 NC^1 circuits. We first need to find a counterpart of the MDDH assumption in NC^1, since the separation assumption $\mathsf{NC}^1 \subsetneq \oplus\mathsf{L}/\mathsf{poly}$ does not directly give us tools in constructing cryptographic schemes. In the work of [21, 1], it is shown that, if $\mathsf{NC}^1 \subsetneq \oplus\mathsf{L}/\mathsf{poly}$ holds, then the following two distributions are indistinguishable for NC^1 circuits:

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

where $n = n(\lambda)$ is some polynomial in security parameter λ, and the randomized sampling algorithms $\mathsf{ZeroSamp}$ and $\mathsf{OneSamp}$ 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 NC^1 implicitly given by Egashira, Wang, and Tanaka [15] for their HPS: Given a matrix $\mathbf{M}^\top$ from D_0 and a random vector $\mathbf{t}$ in

[1] Essentially, the BKP framework used the GS proof for linear equations and replaced the GS commitment with the Pedersen commitment.

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 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 NC^1. We prove that, if the subset membership problem is hard in NC^1, then our MAC is secure for 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 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 NC^1. At the core of our proof system is a new commitment scheme in 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 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 $\mathbf{r} \xleftarrow{\$} \{0\} \times \{0,1\}^{n-1}$, it is possible to extract the forgery from our 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 NC^1 and ABE. Our techniques for IBE can also be viewed as the dual system encryption methodology [35] in 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 [35, 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 (part of) a predicate encoding scheme [36, 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 NC^1 via the rest part of the predicate encoding scheme. Here, the predicate encoding scheme is to construct ABEs in a modular and generic way, in particular, it can generalize the encoding of different ABEs (such as identity-based encryption and inner-product encryption).

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 NC^1 with adaptive soundness in Appendix B. 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 [17, 20]. We leave a detailed treatment of it as an open problem.

1.4 Comparison with the Proceedings Version

This is the full version of the paper appeared at Crypto 2021 [34]. In this full version, we give the full proof for the security of the fine-grained IBE scheme in Section 4.2. Additionally, we give the proof of Theorem 2.13, the definition, constructions, and security proofs of the fine-grained QA-NIZKs (with the comparison to previous and subsequent fine-grained NIZKs [2, 32, 33]), and the instantiations of predicate encodings, in Appendices A to C.

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 $a \xleftarrow{\$} \mathcal{A}(b)$ (respectively, $a = \mathcal{A}(b)$) to denote the random variable outputted by a probabilistic (respectively, deterministic) algorithm $\mathcal{A}$ on input b. By $x \xleftarrow{\$} \mathcal{S}$ 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 $\{1, \cdots, n\}$. By x_i (respectively, x_i) we denote the ith element of a vector $\mathbf{x}$ (respectively, x). By negl we denote an unspecified negligible function.

For a matrix $\mathbf{A} \in \{0,1\}^{n \times t}$ with 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 $\mathrm{Im}(\mathbf{A})$ (i.e., the span of $\mathbf{A}$) and $\mathrm{Ker}(\mathbf{A})$ 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 $\mathbf{y} \notin \mathrm{Im}(\mathbf{A})$, we have $\mathbf{y}^{\top}\mathbf{A}^{\perp} \neq \mathbf{0}$. By $(a_{ij})_{i \in [l], j \in [m]}$ we denote the matrix $\begin{pmatrix} a_{11} \cdots a_{1m} \\ \vdots \ddots \vdots \\ a_{l1} \cdots a_{lm} \end{pmatrix}$. 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

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

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

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

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

Games. We follow [5] to use code-based games for defining and proving security. A game G contains procedures Init and Finalize, and some additional procedures $\mathrm{P}_1, \ldots, \mathrm{P}_n$, 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 G w.r.t. the security parameter λ (denote by G^{a_λ}) if a_λ 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 $\mathsf{G}^{a_\lambda} \Rightarrow d$ to denote that G outputs d after interacting with a_λ, and d is the output of Finalize.

2.1 Function Families

In this section, we recall the definitions of function families, NC^1 circuits, $\mathsf{AC}^0[2]$ circuits, and $\oplus\mathsf{L/poly}$. Note that $\mathsf{AC}^0[2] \subsetneq \mathsf{NC}^1$ [28, 31].

Definition 2.1 (Function Family). A function family is a family of (possibly randomized) functions $\mathcal{F} = \{f_\lambda\}_{\lambda \in \mathbb{N}}$, where for each λ, f_λ has a domain D_λ^f and a range R_λ^f.

Definition 2.2 (NC^1). The class of (non-uniform) 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 λ, f_λ 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 2.3 ($\mathsf{AC}^0[2]$). The class of (non-uniform) $\mathsf{AC}^0[2]$ 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 λ, f_λ 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 $\mathsf{AC}^0[2]$, since it can be done in constant depth with PARITY gates.

Definition 2.4 ($\oplus\mathsf{L/poly}$). $\oplus\mathsf{L/poly}$ 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 λ, there is a non-deterministic Turing machine $\mathcal{M}_\lambda$ such that for each input x with length λ, $\mathcal{M}_\lambda(x)$ 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, $\mathsf{ZeroSamp}$, and $\mathsf{OneSamp}$ in Figure 1. Note that the output of $\mathsf{ZeroSamp}(n)$ is always a matrix of rank $n-1$ and the output of

$\underline{\mathsf{LSamp}(n)}$:
For all $i,j \in [n]$ and $i<j$:
$r_{i,j} \xleftarrow{\$} \{0,1\}$
Return

$$\begin{pmatrix} 1 & r_{1,2} & \cdots & r_{1,n-1} & r_{1,n} \\ 0 & 1 & r_{2,3} & \cdots & r_{2,n} \\ 0 & 0 & \ddots & & \vdots \\ \vdots & \vdots & \ddots & 1 & r_{n-1,n} \\ 0 & \cdots & 0 & 0 & 1 \end{pmatrix}$$

$\underline{\mathsf{RSamp}(n)}$:
For $i = 1, \cdots, n-1$
$r_i \xleftarrow{\$} \{0,1\}$
Return

$$\begin{pmatrix} 1 & & \cdots & 0 & r_1 \\ 0 & 1 & & & r_2 \\ 0 & 0 & \ddots & & \vdots \\ \vdots & \vdots & \ddots & 1 & r_{n-1} \\ 0 & \cdots & 0 & 0 & 1 \end{pmatrix}$$

$\underline{\mathsf{ZeroSamp}(n)}$:
$\mathbf{R}_0 \xleftarrow{\$} \mathsf{LSamp}(n) \in \{0,1\}^{n\times n}$
$\mathbf{R}_1 \xleftarrow{\$} \mathsf{RSamp}(n) \in \{0,1\}^{n\times n}$
Return $\mathbf{R}_0\mathbf{M}_0^n\mathbf{R}_1 \in \{0,1\}^{n\times n}$

$\underline{\mathsf{OneSamp}(n)}$:
$\mathbf{R}_0 \xleftarrow{\$} \mathsf{LSamp}(n)$
$\mathbf{R}_1 \xleftarrow{\$} \mathsf{RSamp}(n)$
Return $\mathbf{R}_0\mathbf{M}_1^n\mathbf{R}_1 \in \{0,1\}^{n\times n}$

Figure 1: Definitions of LSamp, RSamp, $\mathsf{ZeroSamp}$, and $\mathsf{OneSamp}$. $n = n(\lambda)$ is a polynomial in the security parameter λ.

$\mathsf{OneSamp}(n)$ is always a matrix of full rank [13].

We now recall several assumptions and lemmata on $\mathsf{ZeroSamp}$ and $\mathsf{OneSamp}$ given in [13].

Definition 2.5 (Fine-grained matrix linear assumption [13]). There exists a polynomial $n = n(\lambda)$ in the security parameter λ such that for any family $\mathcal{A} = \{a_\lambda\}_{\lambda\in\mathbb{N}}$ in NC^1, we have

$$\begin{aligned} |\Pr[a_\lambda(\mathbf{M}) = 1 \mid \mathbf{M} \xleftarrow{\$} \mathsf{ZeroSamp}(n)] - \\ \Pr[a_\lambda(\mathbf{M}') = 1 \mid \mathbf{M}' \xleftarrow{\$} \mathsf{OneSamp}(n)]| \le \mathsf{negl}(\lambda). \end{aligned}$$

Lemma 2.6 (Lemma 4.3 in [13]). *If $\mathsf{NC}^1 \subsetneq \oplus\mathsf{L/poly}$, then the fine-grained matrix linear assumption holds.*

Remark. Notice that for any polynomial $n = n(\lambda)$, we have $\{f_n\}_{\lambda\in N} \in \mathsf{NC}^1$ iff $\{f_\lambda\}_{\lambda\in N} \in \mathsf{NC}^1$ 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 $\mathsf{ZeroSamp}(\cdot)$ and $\mathsf{OneSamp}(\cdot)$ take as input λ. Moreover, we note that we adopt the stronger notion of the assumption $\mathsf{NC}^1 \subsetneq \oplus\mathsf{L/poly}$ (mentioned in the second paragraph in Remark 3.1 in [13]) for simplicity. Specifically, we assume the existence of functions in $\oplus\mathsf{L/poly}$ that are not computable in NC^1 for all (sufficiently large) security parameter. If we adopt the infinitely-often version of $\mathsf{NC}^1 \subsetneq \oplus\mathsf{L/poly}$, which only assumes that the above hold for an infinitely large number of values of λ, one can see that our primitives achieve security in an infinitely-often sense (see the first paragraph in Remark 3.1 in [13]). The same argument can also be applied to the fine-grained primitives in [15, 32].

The following lemma implies that for a matrix $\mathbf{M}^\top$ sampled by $\mathsf{ZeroSamp}(\lambda)$, 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 2.7 (Lemma 3 in [15]). *For all $\lambda \in \mathbb{N}$ and all $\mathbf{M}^\top \in \mathsf{ZeroSamp}(\lambda)$, it holds that $\mathrm{Ker}(\mathbf{M}^\top) = \{\mathbf{0}, \mathbf{k}\}$ where $\mathbf{k}$ is a vector such that $\mathbf{k} \in \{0,1\}^{\lambda-1} \times \{1\}$.*

Lemma 2.8 (Lemma 4 in [15]). *For all $\lambda \in \mathbb{N}$ and all $\mathbf{M}^\top \in \mathsf{ZeroSamp}(\lambda)$, it holds that $\mathrm{Ker}(\mathbf{M}) = \{\mathbf{0}, \mathbf{k}\}$ where $\mathbf{k}$ is a vector such that $\mathbf{k} \in \{1\} \times \{0,1\}^{\lambda-1}$.*

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

Lemma 2.9 (Lemma 7 in [15]). *For all $\lambda \in \mathbb{N}$, the distributions of $\mathbf{M} + \mathbf{N}^\lambda$ and $\mathbf{M}'$ are identical, where $\mathbf{M}^\top \xleftarrow{\$} \mathsf{ZeroSamp}(\lambda)$ and $\mathbf{M}'^\top \xleftarrow{\$} \mathsf{OneSamp}(\lambda)$.*

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 $\mathbf{Mw}$ for $\mathbf{M}^\top \in \mathsf{ZeroSamp}(\lambda)$.

Lemma 2.10 (Lemma 5 in [15]). *For all $\lambda \in \mathbb{N}$ and all $\mathbf{M}^\top \in \mathsf{ZeroSamp}(\lambda)$, it holds that*

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

Lemma 2.11 *For all $\lambda \in \mathbb{N}$ and all $\mathbf{M}^\top \in \mathsf{ZeroSamp}(\lambda)$, the distributions of $\mathbf{x}$ and $\mathbf{x}'$ are identical, where $\mathbf{w} \xleftarrow{\$} \{0\} \times \{0,1\}^{\lambda-1}$, $\mathbf{w}' \xleftarrow{\$} \{1\} \times \{0,1\}^{\lambda-1}$, $\mathbf{x} = \mathbf{Mw}$, and $\mathbf{x}' = \mathbf{Mw}'$.*

Proof. According to Lemma 2.8, for any $\mathbf{M}^\top \in \mathsf{ZeroSamp}(\lambda)$, there exists $\mathbf{k} \in \mathrm{Ker}(\mathbf{M})$ such that $\mathbf{k} \in \{1\} \times \{0,1\}^{\lambda-1}$. Therefore, the distributions of $(\mathbf{w} + \mathbf{k})$, where $\mathbf{w} \xleftarrow{\$} \{0\} \times \{0,1\}^{\lambda-1}$, and $\mathbf{w}' \xleftarrow{\$} \{1\} \times \{0,1\}^{\lambda-1}$ are identical. Moreover, we have $\mathbf{Mw} = \mathbf{M}(\mathbf{w} + \mathbf{k})$. Hence, the distributions of $\mathbf{Mw}$ and $\mathbf{Mw}'$ are identical, completing the proof of Lemma 2.11. □

Below we recall the a theorem implicitly given in [15] as the subset membership problem for an HPS. Roughly, it shows that for $\mathbf{M}^\top \xleftarrow{\$} \mathsf{ZeroSamp}(\lambda)$, a vector sampled from the span of $\mathbf{M}$ is indistinguishable from one sampled outside the span of $\mathbf{M}$ for any adversary in NC^1. The proof of this theorem is given in Appendix A for completeness.

Definition 2.12 (Fine-grained subset membership problem [15]). Let $\mathsf{SY} = \{\mathsf{SampYes}_\lambda\}_{\lambda \in \mathbb{N}}$ and $\mathsf{SN} = \{\mathsf{SampNo}_\lambda\}_{\lambda \in \mathbb{N}}$ be function families described in Figure 2. For all $\lambda \in \mathbb{N}$, all $\mathbf{M}^\top \in \mathsf{ZeroSamp}(\lambda)$, and all $\mathbf{x} \in \mathsf{SampNo}_\lambda(\mathbf{M})$, we have $\mathbf{x} \in \{0,1\}^\lambda \setminus \mathrm{Im}(\mathbf{M})$, then for $\mathbf{M}^\top \xleftarrow{\$} \mathsf{ZeroSamp}(\lambda)$ and any adversary $\mathcal{A} = \{a_\lambda\}_{\lambda \in \mathbb{N}} \in \mathsf{NC}^1$, we have

$$\begin{aligned} |\Pr[a_\lambda(\mathbf{x}, \mathbf{M}) = 1 \mid \mathbf{x} \xleftarrow{\$} \mathsf{SampYes}_\lambda(\mathbf{M})] - \\ \Pr[a_\lambda(\mathbf{x}, \mathbf{M}) = 1 \mid \mathbf{x} \xleftarrow{\$} \mathsf{SampNo}_\lambda(\mathbf{M})]| \leq \mathsf{negl}(\lambda). \end{aligned}$$

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

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

Theorem 2.13 ([15]). *If $\mathsf{NC}^1 \subsetneq \oplus\mathsf{L}/\mathsf{poly}$, then the fine-grained subset membership problem (see Definition 2.12) holds.*

Remark. Note that the subset membership problem in [15] gives a stronger result additionally showing that the output distributions of $\mathsf{SampYes}_\lambda(\mathbf{M})$ and $\mathsf{SampNo}_\lambda(\mathbf{M})$ are identical to the uniform distributions over $\mathrm{Im}(\mathbf{M})$ and $\{0,1\}^\lambda \setminus \mathrm{Im}(\mathbf{M})$ 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 2.14 (Predicate Encoding [10]). Let $\mathsf{P} = \{\mathsf{p}_\lambda\}_{\lambda\in\mathbb{N}}$ with $\mathsf{p}_\lambda : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ be a predicate, where $\mathcal{X}$ and $\mathcal{Y}$ are polynomial-sized spaces associated with λ. An $\mathcal{C}_1$*-predicate encoding* for P is a function family $\mathsf{PE} = \{\mathsf{rE}_\lambda, \mathsf{kE}_\lambda, \mathsf{sE}_\lambda, \mathsf{sD}_\lambda, \mathsf{rD}_\lambda\}_{\lambda\in\mathbb{N}} \in \mathcal{C}_1$ with

- $\mathsf{rE}_\lambda : \mathcal{Y} \times \{0,1\}^\ell \to \{0,1\}^\eta$,
- $\mathsf{kE}_\lambda : \mathcal{Y} \times \{0,1\} \to \{0,1\}^\eta$,
- $\mathsf{sE}_\lambda : \mathcal{X} \times \{0,1\}^\ell \to \{0,1\}^\zeta$,
- $\mathsf{sD}_\lambda : \mathcal{X} \times \mathcal{Y} \times \{0,1\}^\zeta \to \{0,1\}$,
- $\mathsf{rD}_\lambda : \mathcal{X} \times \mathcal{Y} \times \{0,1\}^\eta \to \{0,1\}$,

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

Linearity is satisfied is for all $\lambda \in \mathbb{N}$ and all $(\mathsf{x}, \mathsf{y}) \in \mathcal{X} \times \mathcal{Y}$, $\mathsf{rE}_\lambda(\mathsf{y}, \cdot)$, $\mathsf{kE}_\lambda(\mathsf{y}, \cdot)$, $\mathsf{sE}_\lambda(\mathsf{x}, \cdot)$, $\mathsf{sD}_\lambda(\mathsf{x}, \mathsf{y}, \cdot)$, and $\mathsf{rD}_\lambda(\mathsf{x}, \mathsf{y}, \cdot)$ are $\{0,1\}$-linear. Namely, for any $\mathsf{y} \in \mathcal{Y}$, any $\mathbf{w}_0, \mathbf{w}_1 \in \{0,1\}^\ell$, and any $c \in \{0,1\}$, we have $\mathsf{rE}_\lambda(\mathsf{y}, \mathbf{w}_0 + \mathbf{w}_1 \cdot c) = \mathsf{rE}_\lambda(\mathsf{y}, \mathbf{w}_0) + \mathsf{rE}_\lambda(\mathbf{w}_1) \cdot c$, and the same argument can be made for kE_λ, sE_λ, sD_λ, and rD_λ.

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

$$\mathsf{rD}_\lambda(\mathsf{x}, \mathsf{y}, \mathsf{rE}_\lambda(\mathsf{y}, \mathsf{w})) = \mathsf{sD}_\lambda(\mathsf{x}, \mathsf{y}, \mathsf{sE}_\lambda(\mathsf{x}, \mathsf{w})) \text{ and } \mathsf{rD}_\lambda(\mathsf{x}, \mathsf{y}, \mathsf{kE}_\lambda(\mathsf{y}, \alpha)) = \alpha.$$

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

$$(\mathsf{x}, \mathsf{y}, \alpha, \mathsf{sE}_\lambda(\mathsf{x}, \mathsf{w}), \mathsf{rE}_\lambda(\mathsf{y}, \mathsf{w}) + \mathsf{kE}_\lambda(\mathsf{y}, \alpha)) \text{ and } (\mathsf{x}, \mathsf{y}, \alpha, \mathsf{sE}_\lambda(\mathsf{x}, \mathsf{w}), \mathsf{rE}_\lambda(\mathsf{y}, \mathsf{w})),$$

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

Intuitively, in a modularly designed attribute-based encryption (ABE) scheme, the attribute value in the user's key is encoded by rE_λ and kE_λ, while that in the ciphertext is encoded by sE_λ. The decryption algorithm uses the associated decoding algorithms rD_λ and sD_λ to decode $(\mathsf{rE}_\lambda, \mathsf{kE}_\lambda)$ and sE_λ respectively. The difference between the decoding results of rD_λ and sD_λ for $(\mathsf{rE}_\lambda, \mathsf{kE}_\lambda)$ and sE_λ is the decoding result of rD_λ for kE_λ only, which is used to yield the session key. These encoding algorithms can be instantiated according to the predicates of different types of ABEs, thus allowing for a modular and generic approach to ABE construction. Namely, different ABE schemes can be constructed by plugging in different encoding algorithms, based on the desired access structure for the scheme.

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

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

for all i, j to denote the matrix

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

The same argument is made for $(\mathsf{kE}_\lambda, \mathsf{sE}_\lambda, \mathsf{sD}_\lambda, \mathsf{rD}_\lambda)$.

Encoding for equality. We now give an example of predicate encoding $\mathsf{PE}_{\mathsf{eq}}$ for equality P_{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 α-reconstruction follows from the fact that $u + \mathbf{x}^\top \mathbf{w} = u + \mathbf{y}^\top \mathbf{w}$ when $\mathbf{x} = \mathbf{y}$, and α-privacy follows from the fact that $u + \mathbf{x}^\top \mathbf{w}$ and $u + \mathbf{y}^\top \mathbf{w}$ are pairwise independent if $\mathbf{x} \neq \mathbf{y}$.

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

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

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 cipher.

Definition 2.15 (Attribute-Based Key Encapsulation). A $\mathcal{C}_1$-*attribute-based key encapsulation (ABKEM) scheme* for a predicate $\mathsf{P} = \{\mathsf{p}_\lambda\}_\lambda$ is a function family $\mathsf{ABKEM} = \{\mathsf{Gen}_\lambda, \mathsf{USKGen}_\lambda, \mathsf{Enc}_\lambda, \mathsf{Dec}_\lambda\}_{\lambda\in\mathbb{N}} \in \mathcal{C}_1$ with the following properties.

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

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

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

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

Definition 2.16 (PR-AT-CPA Security for ABKEM). Let $k(\cdot)$ and $l(\cdot)$ be functions in λ. 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_λ is allowed to make k rounds of adaptive queries to $\textsc{USKGen}(\cdot)$ and each round it queries l inputs, we have

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

where the experiments are defined in Figure 4.

$\underline{\textsc{Init}:}$ $(\mathsf{pk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{Gen}_\lambda$ Return pk $\underline{\textsc{USKGen}(\mathsf{y}):}$ $//k(\lambda) \times l(\lambda)$ queries $\mathcal{Q}_\mathsf{y} \xleftarrow{\$} \mathcal{Q}_\mathsf{y} \cup \{\mathsf{y}\}$ Return $\mathsf{usk}[\mathsf{id}] \xleftarrow{\$} \mathsf{USKGen}_\lambda(\mathsf{sk}, \mathsf{y})$	$\underline{\textsc{Enc}(\mathsf{x}):}$ //one query $(\mathsf{K}^*, \mathsf{ct}^*) \xleftarrow{\$} \mathsf{Enc}_\lambda(\mathsf{pk}, \mathsf{x})$ $\boxed{\mathsf{K}^* \xleftarrow{\$} \mathcal{K}}$ Return $(\mathsf{K}^*, \mathsf{ct}^*)$ $\underline{\textsc{Finalize}(\beta):}$ If $(\mathsf{p}_\lambda(\mathsf{x}, \mathsf{y}) \neq 1$ for all $\mathsf{y} \in \mathcal{Q}_\mathsf{y}$, return β Else return 0

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

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 $\mathsf{NC}^1 \subsetneq \oplus\mathsf{L/poly}$.

3.1 Definitions

The definition of generalized affine MAC is as follows.

Definition 3.1 (Generalized Affine MAC). Let $\mathsf{PE} = \{\mathsf{sE}_\lambda, \mathsf{rE}_\lambda, \mathsf{kE}_\lambda, \mathsf{sD}_\lambda, \mathsf{rD}_\lambda\}_{\lambda\in\mathbb{N}} \in \mathcal{C}_1$ be a predicate encoding for $\mathsf{P} = \{\mathsf{p}_\lambda\}_{\lambda\in\mathbb{N}}$, where $\mathsf{rE}_\lambda : \mathcal{Y} \times \{0,1\}^\ell \to \{0,1\}^\eta$, $\mathsf{kE}_\lambda : \mathcal{Y} \times \{0,1\} \to \{0,1\}^\eta$, and $\mathsf{sE}_\lambda : \mathcal{X} \times \{0,1\}^\ell \to \{0,1\}^\varsigma$.

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

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

$$\mathbf{t} \xleftarrow{\$} \mathsf{SampYes}_\lambda(\mathbf{B}) \tag{1}$$
$$\mathbf{u} = \mathsf{rE}_\lambda(\mathsf{m}, \mathbf{X}^\top \mathbf{t}) + \mathsf{kE}_\lambda(\mathsf{m}, x') \in \{0,1\}^\eta. \tag{2}$$

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

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

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

Definition 3.2 (PR-CMA Security). Let $k = k(\lambda)$ and $l = l(\lambda)$ be polynomials in λ. $\mathsf{MAC}_{\mathsf{GA}}$ 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_λ is allowed to make k rounds of adaptive queries to $\textsc{Eval}(\cdot)$ and each round it queries l inputs, we have

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

where the experiments are defined in Figure 5.

$\underline{\textsc{Init}:}$ $\mathsf{sk}_{\mathsf{MAC}} = (\mathbf{B}, \mathbf{X}, x') \xleftarrow{\$} \mathsf{Gen}_{\mathsf{MAC}\lambda}(\mathsf{par})$ Return ε $\underline{\textsc{Eval}(\mathsf{m}):}$ // $k(\lambda) \times \ell(\lambda)$ queries $\mathcal{Q}_{\mathsf{m}} = \mathcal{Q}_{\mathsf{m}} \cup \{\mathsf{m}\}$ Return $(\mathbf{t}, \mathbf{u}) \xleftarrow{\$} \mathsf{Tag}_\lambda(\mathsf{sk}_{\mathsf{MAC}}, \mathsf{m})$	$\underline{\textsc{Chal}(\mathsf{m}^*):}$ //one query $\mathbf{h}_0 = \mathsf{sE}_\lambda(\mathsf{m}^*, \mathbf{X}^\top) \in \{0,1\}^{\zeta \times \lambda}$ $h_1 = x' \in \{0,1\}$ $\boxed{h_1 \xleftarrow{\$} \{0,1\}}$ Return $(\mathbf{h}_0, h_1)$ $\underline{\textsc{Finalize}(\beta \in \{0,1\}):}$ If $\mathsf{p}_\lambda(\mathsf{m}^*, \mathsf{m}) \neq 1$ for all $\mathsf{m} \in \mathcal{Q}_{\mathsf{m}}$, return β Else return 0

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

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

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

$$u = \mathbf{x}_0^\top \mathbf{t} + \sum_{i=1}^{n} \mathsf{m}_i \mathbf{x}_i^\top \mathbf{t} + x' \in \{0,1\} \tag{3}$$

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

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

in Figure 5, i.e., the predicate encoding is the one for equality (see Figure 3), the above definition becomes 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 $\mathsf{SampYes}_\lambda$. We give the definition as below.

Definition 3.3 (Affine MAC [5]). A *Generalized affine MAC* for the predicate P_{eq} and encoding $\mathsf{PE}_{\mathsf{eq}}$ defined as in Figure 3 is said to be an affine MAC.

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

Figure 6: Definition of $\mathsf{MAC}_{\mathsf{GA}} = \{\mathsf{Gen}_{\mathsf{MAC}\lambda}, \mathsf{Tag}_\lambda, \mathsf{Ver}_{\mathsf{MAC}\lambda}\}_{\lambda \in \mathbb{N}}$.

3.2 Construction

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

Theorem 3.4 *If $\mathsf{NC}^1 \subsetneq \oplus\mathsf{L}/\mathsf{poly}$ and $\mathsf{PE} = \{\mathsf{sE}_\lambda, \mathsf{rE}_\lambda, \mathsf{kE}_\lambda, \mathsf{sD}_\lambda, \mathsf{rD}_\lambda\}_{\lambda \in \mathbb{N}} \in \mathsf{AC}^0[2]$ is a predicate encoding, where $\mathsf{rE}_\lambda : \mathcal{Y} \times \{0,1\}^\ell \to \{0,1\}^\eta$, $\mathsf{kE}_\lambda : \mathcal{Y} \times \{0,1\} \to \{0,1\}^\eta$, and $\mathsf{sE}_\lambda : \mathcal{X} \times \{0,1\}^\ell \to \{0,1\}^\zeta$, then $\mathsf{MAC}_{\mathsf{GA}}$ is an $\mathsf{AC}^0[2]$-generalized affine MAC that is NC^1-(k,l)-PR-CMA secure, where k is any constant and $l = l(\lambda)$ is any polynomial in λ.*

$\underline{\textsc{Init}}$: // Games G_0-G_2	$\underline{\textsc{Eval}(\mathsf{m})}$: // Game G_0
$\mathbf{B}^\top \xleftarrow{\$} \mathsf{ZeroSamp}(\lambda)$, $x' \xleftarrow{\$} \{0,1\}$	$\mathcal{Q}_\mathsf{m} = \mathcal{Q}_\mathsf{m} \cup \{\mathsf{m}\}$
For $\mathbf{X} \xleftarrow{\$} \{0,1\}^{\lambda \times \ell}$	$\mathbf{t} \xleftarrow{\$} \mathsf{SampYes}_\lambda(\mathbf{B})$
Return ε	$\mathbf{u} = \mathsf{rE}_\lambda(\mathsf{m}, \mathbf{X}^\top \mathbf{t}) + \mathsf{kE}_\lambda(\mathsf{m}, x') \in \{0,1\}^\eta$
	Return $(\mathbf{t}, \mathbf{u})$
$\underline{\textsc{Chal}(\mathsf{m}^* \in \mathcal{X})}$: //Games G_0-$\mathsf{G}_{1,Q+1}$, $\boxed{\mathsf{G}_2}$	$\underline{\textsc{Eval}(\mathsf{m})}$: // Games $\mathsf{G}_{1,i}$, $\boxed{\mathsf{G}'_{1,i}}$
$\mathbf{h}_0 = \mathsf{sE}_\lambda(\mathsf{m}^*, \mathbf{X}^\top) \in \{0,1\}^{\zeta \times \lambda}$	$\mathcal{Q}_\mathsf{m} = \mathcal{Q}_\mathsf{m} \cup \{\mathsf{m}\}$ // Let m be the c-th query
$h_1 = x' \in \{0,1\}$	$(1 \le c \le k \cdot l)$
$\boxed{h_1 \xleftarrow{\$} \{0,1\}}$	If $c < i$ then
Return $(\mathbf{h}_0, h_1)$	$\mathbf{t} \xleftarrow{\$} \mathsf{SampNo}_\lambda(\mathbf{B})$
	$\mathbf{u} = \mathsf{rE}_\lambda(\mathsf{m}, \mathbf{X}^\top \mathbf{t}) \in \{0,1\}^\eta$
$\underline{\textsc{Finalize}(\beta \in \{0,1\})}$: // Games G_0-G_2	If $c > i$ then
If $\mathsf{p}_\lambda(\mathsf{m}^*, \mathsf{m}) \neq 1$ for all $\mathsf{m} \in \mathcal{Q}_\mathsf{m}$	$\mathbf{t} \xleftarrow{\$} \mathsf{SampYes}_\lambda(\mathbf{B})$
return β	$\mathbf{u} = \mathsf{rE}_\lambda(\mathsf{m}, \mathbf{X}^\top \mathbf{t}) + \mathsf{kE}_\lambda(\mathsf{m}, x') \in \{0,1\}^\eta$
Else return 0	If $c = i$ then
	$\mathbf{t} \xleftarrow{\$} \mathsf{SampYes}_\lambda(\mathbf{B})$
$\underline{\textsc{Eval}(\mathsf{m})}$: //Game G_2	$\boxed{\mathbf{t} \xleftarrow{\$} \mathsf{SampNo}_\lambda(\mathbf{B})}$
$\mathcal{Q}_\mathsf{m} = \mathcal{Q}_\mathsf{m} \cup \{\mathsf{m}\}$	$\mathbf{u} = \mathsf{rE}_\lambda(\mathsf{m}, \mathbf{X}^\top \mathbf{t}) + \mathsf{kE}_\lambda(\mathsf{m}, x') \in \{0,1\}^\eta$
$\mathbf{t} \xleftarrow{\$} \mathsf{SampNo}_\lambda(\mathbf{B})$	Return $(\mathbf{t}, \mathbf{u})$
$\mathbf{u} = \mathsf{rE}_\lambda(\mathsf{m}, \mathbf{X}^\top \mathbf{t}) \in \{0,1\}^\eta$	
Return $(\mathbf{t}, \mathbf{u})$	

Figure 7: Games $\mathsf{G}_0, (\mathsf{G}_{1,i}, \mathsf{G}'_{1,i})_{1 \le i \le k \cdot l}, \mathsf{G}_{1,k \cdot l+1}, \mathsf{G}_2$ for the proof of Theorem 3.4.

Proof. First, we note that $(\{\mathsf{Gen}_{\mathsf{MAC}\lambda}\}_{\lambda \in \mathbb{N}}, \{\mathsf{Tag}_\lambda\}_{\lambda \in \mathbb{N}}, \{\mathsf{Ver}_{\mathsf{MAC}\lambda}\}_{\lambda \in \mathbb{N}})$ are computable in $\mathsf{AC}^0[2]$, 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 $\mathsf{MAC}_{\mathsf{GA}}$ satisfies correctness.

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

Let $\mathcal{A} = \{a_\lambda\}_{\lambda \in \mathbb{N}} \in \mathsf{NC}^1$ be any adversary against the PR-CMA-security of $\mathsf{MAC}_{\mathsf{GA}}$. Game G_0 is the real attack game. In games $\mathsf{G}_{1,i}$, the first $i-1$ queries to the $\textsc{Eval}$ oracle are answered with $(\mathbf{t}, \mathbf{u})$, where $\mathbf{t} \xleftarrow{\$} \mathsf{SampNo}_\lambda(\mathbf{B})$ and $\mathbf{u}$ contains no information on $\mathsf{kE}_\lambda(\mathsf{m}, x')$, and the remaining are answered as in

the real scheme. To interpolate between $\mathsf{G}_{1,i}$ and $\mathsf{G}_{1,i+1}$, we also define $\mathsf{G}'_{1,i}$, which answers the i-th query to EVAL by picking $\mathbf{t} \xleftarrow{\$} \mathsf{SampNo}_\lambda(\mathbf{B})$. By definition, we have $\mathsf{G}_0 = \mathsf{G}_{1,1}$.

Lemma 3.5 $\Pr[\mathsf{PR\text{-}CMA}^{a_\lambda}_{\mathsf{real}} \Rightarrow 1] = \Pr[\mathsf{G}^{a_\lambda}_0 \Rightarrow 1] = \Pr[\mathsf{G}^{a_\lambda}_{1,1} \Rightarrow 1]$.

Lemma 3.6 *There exists an adversary $\mathcal{B}_{1,i} = \{b^{1,i}_\lambda\}_{\lambda\in\mathbb{N}} \in \mathsf{NC}^1$ such that $b^{1,i}_\lambda$ breaks the fine-grained subset membership problem (see Definition 2.12), which holds under $\mathsf{NC}^1 \subsetneq \oplus\mathsf{L}/\mathsf{poly}$ according to Theorem 2.13, with probability*

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

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

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

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

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

Lemma 3.7 $\Pr[\mathsf{G}^{a_\lambda}_{1,i+1} \Rightarrow 1] = \Pr[\mathsf{G}'^{a_\lambda}_{1,i} \Rightarrow 1]$.

Proof. Let m be the i-th query to EVAL such that $\mathsf{p}_\lambda(\mathsf{m}^*, \mathsf{m}) \neq 1$ and let $(\mathbf{t}, \mathbf{u})$ be its tag. We have $\mathbf{t} \notin \mathrm{Im}(\mathbf{B})$ due to Theorem 2.13. We use an information-theoretic argument to show that in $\mathsf{G}'_{1,i}$, $\mathbf{u}$ does not reveal any information on x'. Information-theoretically, a_λ 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_λ information-theoretically obtains the distribution of

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

This distribution is identical to the distribution of

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

since the distribution of

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

and

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

are identical due to the α-privacy of PE, completing this part of proof. □

Lemma 3.8 $\Pr[\mathsf{G}^{a_\lambda}_2 \Rightarrow 1] = \Pr[\mathsf{G}^{a_\lambda}_{1,k\cdot l+1} \Rightarrow 1]$.

Proof. Note that a_λ can ask at most $k \cdot l$-many EVAL queries. In both $\mathsf{G}_{1,k\cdot l+1}$ and G_2, all the answers of EVAL are independent of x'. Hence, h_1 from $\mathsf{G}_{1,k\cdot l+1}$ is uniform in the view of a_λ. □

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 3.9 *There exists an adversary $\mathcal{B}_2 = \{b^2_\lambda\}_{\lambda\in\mathbb{N}} \in \mathsf{NC}^1$ such that b^2_λ breaks the fine-grained subset membership problem with probability at least*

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

$\underline{\textsc{Init}:}$ // Games H_0-H_2
$\mathbf{B}^\top \xleftarrow{\$} \mathsf{ZeroSamp}(\lambda)$; $x' \xleftarrow{\$} \{0,1\}$
$\mathbf{X} \xleftarrow{\$} \{0,1\}^{\lambda\times\ell}$
Return ε
 	
$\underline{\textsc{Chal}(\mathsf{m}^*):}$ //Games H_0-H_2
$\mathbf{h}_0 = \mathsf{sE}_\lambda(\mathsf{m}^*, \mathbf{X}^\top) \in \{0,1\}^{\zeta\times\lambda}$
$h_1 \xleftarrow{\$} \{0,1\}$
Return $(\mathbf{h}_0, h_1)$
 	
$\underline{\textsc{Finalize}(\beta \in \{0,1\}):}$ // Games H_0-H_2
If $\mathsf{p}_\lambda(\mathsf{m}^*, \mathsf{m}) \neq 1$ for all $\mathsf{y} \in \mathcal{Q}_\mathsf{m}$
return β
Else return 0
 	
$\underline{\textsc{Eval}(\mathsf{m}):}$ //Game H_0
$\mathcal{Q}_\mathsf{m} = \mathcal{Q}_\mathsf{m} \cup \{\mathsf{m}\}$
$\mathbf{t} \xleftarrow{\$} \mathsf{SampNo}_\lambda(\mathbf{B})$
$\mathbf{u} = \mathsf{rE}_\lambda(\mathsf{m}, \mathbf{X}^\top\mathbf{t}) \in \{0,1\}^\eta$
Return $(\mathbf{t}, \mathbf{u})$
	$\underline{\textsc{Eval}(\mathsf{m}):}$ // Games $\mathsf{H}_{1,i}$, $\boxed{\mathsf{H}'_{1,i}}$
	$\mathcal{Q}_\mathsf{m} = \mathcal{Q}_\mathsf{m} \cup \{\mathsf{m}\}$ // Let m be the c-th query
	$(1 \le c \le k \cdot l)$
	If $c > i$ then
	$\mathbf{t} \xleftarrow{\$} \mathsf{SampNo}_\lambda(\mathbf{B})$
	$\mathbf{u} = \mathsf{rE}_\lambda(\mathsf{m}, \mathbf{X}^\top\mathbf{t}) \in \{0,1\}^\eta$
	If $c < i$ then
	$\mathbf{t} \xleftarrow{\$} \mathsf{SampYes}_\lambda(\mathbf{B})$
	$\mathbf{u} = \mathsf{rE}_\lambda(\mathsf{m}, \mathbf{X}^\top\mathbf{t}) + \mathsf{kE}_\lambda(\mathsf{m}, x') \in \{0,1\}^\eta$
	If $c = i$ then
	$\mathbf{t} \xleftarrow{\$} \mathsf{SampNo}_\lambda(\mathbf{B})$
	$\boxed{\mathbf{t} \xleftarrow{\$} \mathsf{SampYes}_\lambda(\mathbf{B})}$
	$\mathbf{u} = \mathsf{rE}_\lambda(\mathsf{m})\mathbf{X}^\top\mathbf{t} + \mathsf{kE}_\lambda(\mathsf{m}, x') \in \{0,1\}^\eta$
	Return $(\mathbf{t}, \mathbf{u})$
	
	$\underline{\textsc{Eval}(\mathsf{m}):}$ // Game H_2
	$\mathcal{Q}_\mathsf{m} = \mathcal{Q}_\mathsf{m} \cup \{\mathsf{m}\}$
	$\mathbf{t} \xleftarrow{\$} \mathsf{SampYes}_\lambda(\mathbf{B})$
	$\mathbf{u} = \mathsf{rE}_\lambda(\mathsf{m}, \mathbf{X}^\top\mathbf{t}) + \mathsf{kE}_\lambda(\mathsf{m}, x') \in \{0,1\}^\eta$
	Return $(\mathbf{t}, \mathbf{u})$

Figure 8: Games $\mathsf{H}_0, (\mathsf{H}_{1,i}, \mathsf{H}'_{1,i})_{1\le i\le k\cdot l}, \mathsf{H}_{1,k\cdot l+1}, \mathsf{H}_2$ for the proof of Lemma 3.9.

Putting all above together, Theorem 3.4 immediately follows. □

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 $\mathsf{MAC} = \{\mathsf{Gen}_{\mathsf{MAC}\lambda}, \mathsf{Tag}_\lambda, \mathsf{Ver}_{\mathsf{MAC}\lambda}\}_{\lambda\in\mathbb{N}}$ as in Figure 9 for message space $\{0,1\}^\ell$, which will be used to construct an IBE scheme in NC^1 later. Formally, we have the following corollary derived from Theorem 3.4.

$\underline{\mathsf{Gen}_{\mathsf{MAC}\lambda}(\mathsf{par}):}$ $\mathbf{B}^\top \xleftarrow{\$} \mathsf{ZeroSamp}(\lambda)$ $\mathbf{x}_0, \ldots, \mathbf{x}_\ell \xleftarrow{\$} \{0,1\}^\lambda$ $x' \xleftarrow{\$} \{0,1\}$ Return $\mathsf{sk}_\mathsf{MAC} = (\mathbf{B}, \mathbf{x}_0, \ldots, \mathbf{x}_\ell, x'_0)$	$\underline{\mathsf{Tag}_\lambda(\mathsf{sk}_\mathsf{MAC}, \mathsf{m} \in \{0,1\}^\ell):}$ $\mathbf{t} \xleftarrow{\$} \mathsf{SampYes}_\lambda(\mathbf{B})$ $u = (\mathbf{x}_0^\top + \sum_{i=1}^\ell \mathsf{m}_i \cdot \mathbf{x}_i^\top)\mathbf{t} + x' \in \{0,1\}$ Return $\tau = (\mathbf{t}, u)$ $\underline{\mathsf{Ver}_{\mathsf{MAC}\lambda}(\mathsf{sk}_\mathsf{MAC}, \tau, \mathsf{m}):}$ If $u = (\mathbf{x}_0^\top + \sum_{i=1}^\ell \mathsf{m}_i \cdot \mathbf{x}_i^\top)\mathbf{t} + x'$ return 1 Else return 0

Figure 9: Definition of $\mathsf{MAC} = \{\mathsf{Gen}_{\mathsf{MAC}\lambda}, \mathsf{Tag}_\lambda, \mathsf{Ver}_{\mathsf{MAC}\lambda}\}_{\lambda\in\mathbb{N}}$.

Corollary 3.10 *If $\mathsf{NC}^1 \subsetneq \oplus\mathsf{L/poly}$, then MAC is an $\mathsf{AC}^0[2]$-affine MAC that is NC^1-(k,l)-PR-CMA secure, where k is any constant and $l = l(\lambda)$ is any polynomial in λ.*

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.

4.1 Definition

We now give the definition of fine-grained IBKEM, which is a special case of fine-grained ABKEM (see Definition 2.15) where the boolean predicate is restricted to be the equality predicate.

Definition 4.1 (Identity-Based Key Encapsulation). A $\mathcal{C}_1$-*identity key encapsulation (IBKEM) scheme* is a function family $\mathsf{IBKEM} = \{\mathsf{Gen}_\lambda, \mathsf{USKGen}_\lambda, \mathsf{Enc}_\lambda, \mathsf{Dec}_\lambda\}_{\lambda\in\mathbb{N}} \in \mathcal{C}_1$ with the following properties.

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

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

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

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

Definition 4.2 (PR-ID-CPA Security for IBKEM). Let $k(\cdot)$ and $l(\cdot)$ be functions in λ. 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_λ is allowed to make k rounds of adaptive queries to USKGen$(\cdot)$ and each round it queries l inputs, we have

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

where the experiments are defined in Figure 10.

Procedure Init:	**Procedure** Enc(id^*):
$(\mathsf{pk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{Gen}_\lambda$	//one query
Return pk	$(\mathsf{K}^*, \mathsf{ct}^*) \xleftarrow{\$} \mathsf{Enc}_\lambda(\mathsf{pk}, \mathsf{id}^*)$
	$\boxed{\mathsf{K}^* \xleftarrow{\$} \mathcal{K}}$
Procedure USKGen(id):	Return $(\mathsf{K}^*, \mathsf{ct}^*)$
$//k(\lambda) \times l(\lambda)$ queries	
$\mathcal{Q}_{\mathsf{id}} \xleftarrow{\$} \mathcal{Q}_{\mathsf{id}} \cup \{\mathsf{id}\}$	**Procedure** Finalize(β):
Return $\mathsf{usk}[\mathsf{id}] \xleftarrow{\$} \mathsf{USKGen}_\lambda(\mathsf{sk}, \mathsf{id})$	Return $(\mathsf{id}^* \notin \mathcal{Q}_{\mathsf{id}}) \wedge \beta$

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

4.2 Construction

Let $\mathsf{MAC} = \{\mathsf{Gen}_{\mathsf{MAC}\lambda}, \mathsf{Tag}_\lambda, \mathsf{Ver}_{\mathsf{MAC}\lambda}\}_{\lambda\in\mathbb{N}} \in \mathsf{NC}^1$ be an affine MAC over $\{0,1\}^\lambda$ with message space $\mathcal{ID}$ in Figure 9. Our IBKEM $\mathsf{IBKEM} = \{\mathsf{Gen}_\lambda, \mathsf{USKGen}_\lambda, \mathsf{Enc}_\lambda, \mathsf{Dec}_\lambda\}_{\lambda\in\mathbb{N}}$ for key-space $\mathcal{K} = \{0,1\}$ and identity space $\{0,1\}^\ell$ is defined as in Figure 11. [2]

Theorem 4.3 *Under the assumption* $\mathsf{NC}^1 \subsetneq \oplus\mathsf{L/poly}$ *and the* NC^1-(k, l)-PR-CMA *security of* MAC*, where k is any constant and $l = l(\lambda)$ is any polynomial in λ,* IBKEM *is an* $\mathsf{AC}^0[2]$-*IBKEM that is* NC^1-(k, l)-PR-ID-CPA *secure against* NC^1.

[2] The IBKEM can be straightforwardly extended to one with large key space as we will discuss later in this section.

$\underline{\mathsf{Gen}_\lambda}$:
$\mathbf{A}^\top \xleftarrow{\$} \mathsf{ZeroSamp}(\lambda)$
$\mathsf{sk}_{\mathsf{MAC}} = (\mathbf{B}, \mathbf{x}_0, \ldots, \mathbf{x}_\ell, x') \xleftarrow{\$} \mathsf{Gen}_{\mathsf{MAC}\lambda}(\mathsf{par})$
For $i = 0, \ldots, \ell$:
  $\mathbf{Y}_i \xleftarrow{\$} \{0,1\}^{(\lambda-1)\times\lambda}$
  $\mathbf{Z}_i = (\mathbf{Y}_i^\top || \mathbf{x}_i)\mathbf{A} \in \{0,1\}^{\lambda\times\lambda}$
$\mathbf{y}' \xleftarrow{\$} \{0,1\}^{\lambda-1}$
$\mathbf{z}' = (\mathbf{y}'^\top || x')\mathbf{A} \in \{0,1\}^{1\times\lambda}$
$\mathsf{pk} = (\mathbf{A}, (\mathbf{Z}_i)_{0\le i\le \ell}, \mathbf{z}')$
$\mathsf{sk} = (\mathsf{sk}_{\mathsf{MAC}}, (\mathbf{Y}_i)_{0\le i\le \ell}, \mathbf{y}')$
Return $(\mathsf{pk}, \mathsf{sk})$

$\underline{\mathsf{USKGen}_\lambda(\mathsf{sk}, \mathsf{id} \in \{0,1\}^\ell)}$:
$(\mathbf{t}, u) \xleftarrow{\$} \mathsf{Tag}_\lambda(\mathsf{sk}_{\mathsf{MAC}}, \mathsf{id})$
$\mathbf{v} = \mathbf{t}^\top(\mathbf{Y}_0^\top + \sum_{i=1}^{\ell} \mathsf{id}_i \odot \mathbf{Y}_i^\top) + \mathbf{y}'^\top \in \{0,1\}^{1\times(\lambda-1)}$
Return $\mathsf{usk}[\mathsf{id}] = (\mathbf{t}, u, \mathbf{v})$

$\underline{\mathsf{Enc}_\lambda(\mathsf{pk}, \mathsf{id})}$:
$\mathbf{r} \xleftarrow{\$} \{0\} \times \{0,1\}^{\lambda-1}$
$\mathbf{c}_0 = \mathbf{A}\mathbf{r} \in \{0,1\}^\lambda$
$\mathbf{c}_1 = (\mathbf{Z}_0 + \sum_{i=1}^{\ell} \mathsf{id}_i \odot \mathbf{Z}_i)\mathbf{r} \in \{0,1\}^\lambda$
$\mathsf{K} = \mathbf{z}' \cdot \mathbf{r} \in \{0,1\}$.
Return K and $\mathsf{ct} = (\mathbf{c}_0, \mathbf{c}_1)$

$\underline{\mathsf{Dec}_\lambda(\mathsf{usk}[\mathsf{id}], \mathsf{id}, \mathsf{ct})}$:
Parse $\mathsf{usk}[\mathsf{id}] = (\mathbf{t}, u, \mathbf{v})$
Parse $\mathsf{ct} = (\mathbf{c}_0, \mathbf{c}_1) \in \{0,1\}^\lambda \times \{0,1\}^\lambda$
$\mathsf{K} = (\mathbf{v}|u)\mathbf{c}_0 - \mathbf{t}^\top \mathbf{c}_1$
Return K

Figure 11: Definition of our $\mathsf{IBKEM} = \{\mathsf{Gen}_\lambda, \mathsf{USKGen}_\lambda, \mathsf{Enc}_\lambda, \mathsf{Dec}_\lambda\}_{\lambda\in\mathbb{N}}$ with identity space $\{0,1\}^\ell$ and key space $\{0,1\}$. id_i denotes the ith bit of id for all $i \in [\ell]$.

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

Correctness follows from the fact that by Equation (3) in Section 3.1, we have

$$(\mathbf{v}||u)\mathbf{c}_0 = (\mathbf{t}^\top(\mathbf{Y}_0^\top + \sum_{i=1}^{\ell} \mathsf{id}_i \odot \mathbf{Y}_i^\top) + \mathbf{y}'^\top || \mathbf{t}^\top(\mathbf{x}_0 + \sum_{i=1}^{\ell} \mathsf{id}_i \odot \mathbf{x}_i) + x')\mathbf{A}\mathbf{r}$$

$$\mathbf{t}^\top \mathbf{c}_1 = \mathbf{t}^\top(\mathbf{Y}_0^\top || \mathbf{x}_0 + \sum_{i=1}^{\ell} \mathsf{id}_i \odot (\mathbf{Y}_i^\top || \mathbf{x}_i))\mathbf{A}\mathbf{r}$$

and the difference of the two elements yields $\mathsf{K} = (\mathbf{y}'^\top || x')\mathbf{A}\mathbf{r} = \mathbf{z}' \cdot \mathbf{r}$.

Let $\mathcal{A} = \{a_\lambda\}_{\lambda\in\mathbb{N}}$ be any adversary against the NC^1-(k,l)-PR-ID-CPA security of IBKEM. We now prove the NC^1-(k,l)-PR-ID-CPA security by defining a sequence of games G_0-G_6 as in Figure 12. Roughly, in the first four games, we show how to extract a challenge token for MAC 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_λ reveal no useful information other than the tags and token for MAC.

Lemma 4.4 $\Pr[\mathsf{PR\text{-}ID\text{-}CPA}^{a_\lambda}_{\mathsf{real}} \Rightarrow 1] = \Pr[\mathsf{G}_1^{a_\lambda} \Rightarrow 1] = \Pr[\mathsf{G}_0^{a_\lambda} \Rightarrow 1]$.

Proof. G_0 is the real attack game. In game G_1, we change the simulation of $\mathbf{c}_0^*$, $\mathbf{c}_1^*$ and K^* in $\textsc{Enc}(\mathsf{id}^*)$ 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

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

the view of a_λ in G_1 is identical to its view in G_0, completing this part of proof. □

Lemma 4.5 *There exists an adversary $\mathcal{B}_1 = \{b_\lambda^1\}_{\lambda\in\mathbb{N}}$ such that b_λ^1 breaks the fine-grained matrix linear assumption (see Definition 2.5), which holds under $\mathsf{NC}^1 \subsetneq \oplus\mathsf{L}/\mathsf{poly}$ according to Lemma 2.6, with advantage*

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

$\underline{\textsc{Init}}$: //Games G_0-G_1, G_2-G_3, G_4, G_5-G_6

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

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

$\mathsf{sk}_{\mathsf{MAC}} = (\mathbf{B}, \mathbf{x}_0, \ldots, \mathbf{x}_\ell, x') \xleftarrow{\$} \mathsf{Gen}_{\mathsf{MAC}\lambda}(\mathcal{G})$

For $i = 0, \ldots, \ell$:

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

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

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

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

$\mathsf{pk} = (\mathbf{A}, (\mathbf{Z}_i)_{0\le i\le \ell}, \mathbf{z}')$

$\mathsf{sk} = (\mathsf{sk}_{\mathsf{MAC}}, (\mathbf{Y}_i)_{0\le i\le \ell}, \mathbf{y}')$

Return pk

$\underline{\textsc{Finalize}(\beta)}$: //Games G_0-G_6

Return $(\mathsf{id}^* \notin \mathcal{Q}_{\mathsf{id}}) \wedge \beta$

$\underline{\textsc{USKGen}(\mathsf{id})}$: //Games G_0-G_4, G_5-G_6

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

$\mathbf{v} = \mathbf{t}^\top(\mathbf{Y}_0^\top + \sum_{i=1}^\ell \mathsf{id}_i \odot \mathbf{Y}_i^\top) + \mathbf{y}'^\top \in \{0,1\}^{1\times(\lambda-1)}$

$\mathbf{v} = \mathbf{t}^\top(\mathbf{D}_0 + \sum_{i=1}^\ell \mathsf{id}_i \odot \mathbf{D}_i) + \mathbf{d}' - u \cdot \widetilde{\mathbf{r}}^\top \in \{0,1\}^{1\times(\lambda-1)}$

$\mathsf{usk}[\mathsf{id}] = (\mathbf{t}, u, \mathbf{v})$

Return $\mathsf{usk}[\mathsf{id}]$

$\underline{\textsc{Enc}(\mathsf{id}^*)}$: //Games G_0, G_1-G_4, G_3-G_4, G_5, G_6

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

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

$\mathbf{c}_1^* = \mathbf{Z}_0\mathbf{r} + \sum_{i=1}^\ell \mathsf{id}_i^* \odot \mathbf{Z}_i\mathbf{r} \in \{0,1\}^\lambda$

$\mathbf{c}_1^* = (\mathbf{Y}_0^\top \mid \mathbf{x}_0)(\mathbf{A} + \mathbf{N}^\lambda)\mathbf{r} + \sum_{i=1}^\ell \mathsf{id}_i^* \odot (\mathbf{Y}_i^\top \mid \mathbf{x}_i)(\mathbf{A} + \mathbf{N}^\lambda)\mathbf{r}$

$\mathbf{c}_1^* = \mathbf{Z}_0\mathbf{r} + \mathbf{x}_0 + \sum_{i=1}^\ell \mathsf{id}_i^* \odot (\mathbf{Z}_i\mathbf{r} + \mathbf{x}_i)$

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

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

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

Figure 12: Games G_0-G_6 for the proof of Theorem 4.3.

Proof. G_1 and G_2 only differ in the distribution of $\mathbf{A}$, namely, $\mathbf{A}^\top \xleftarrow{\$} \mathsf{ZeroSamp}(\lambda)$ or $\mathbf{A}^\top \xleftarrow{\$} \mathsf{OneSamp}(\lambda)$, and we build the distinguisher b^1_λ as follows.

b^1_λ runs in exactly the same way as the challenger of G_1 except that in Init, instead of generating $\mathbf{A}$ by itself, it takes as input $\mathbf{A}^\top$ generated as $\mathbf{A}^\top \xleftarrow{\$} \mathsf{ZeroSamp}(\lambda)$ or $\mathbf{A}^\top \xleftarrow{\$} \mathsf{OneSamp}(\lambda)$ from its own challenger. When a_λ outputs β, b^1_λ outputs β as well if id^* was not queried to USKGen. Otherwise, b^1_λ outputs 0.

If $\mathbf{A}$ is generated as $\mathbf{A}^\top \xleftarrow{\$} \mathsf{ZeroSamp}(\lambda)$ (respectively, $\mathbf{A}^\top \xleftarrow{\$} \mathsf{OneSamp}(\lambda)$), the view of a_λ is the same as its view in G_1 (respectively, G_2). Hence, the probability that b^1_λ breaks the fine-grained matrix linear assumption is

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

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

Lemma 4.6 $\Pr[\mathsf{G}_3^{a_\lambda} \Rightarrow 1] = \Pr[\mathsf{G}_2^{a_\lambda} \Rightarrow 1]$.

Proof. In this game, we sample $\mathbf{r}$ in Enc(id^*) as $\mathbf{r} \xleftarrow{\$} \{1\} \times \{0,1\}^{\lambda-1}$ instead of $\mathbf{r} \xleftarrow{\$} \{0\} \times \{0,1\}^{\lambda-1}$. According to Lemma 2.9, the distribution of $\mathbf{A}+\mathbf{N}^\lambda$ in G_2 and G_3 is identical to that of a matrix sampled from $\mathsf{ZeroSamp}$. Then this lemma follows from Lemma 2.11 immediately. □

Lemma 4.7 *There exists an adversary $\mathcal{B}_2 = \{b^2_\lambda\}_{\lambda\in\mathbb{N}} \in \mathsf{NC}^1$ such that b^2_λ breaks the fine-grained matrix linear assumption with advantage*

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

Proof. G_3 and G_4 only differ in the distribution of $\mathbf{A}$, namely, $\mathbf{A}^\top \xleftarrow{\$} \mathsf{OneSamp}(\lambda)$ or $\mathbf{A}^\top \xleftarrow{\$} \mathsf{ZeroSamp}(\lambda)$, and we build the distinguisher b^2_λ as follows.

b^2_λ runs in exactly the same way as the challenger of G_3 except that in Init, instead of generating $\mathbf{A}$ by itself, it takes as input $\mathbf{A}^\top$ generated as $\mathbf{A}^\top \xleftarrow{\$} \mathsf{ZeroSamp}(\lambda)$ or $\mathbf{A}^\top \xleftarrow{\$} \mathsf{OneSamp}(\lambda)$ from its own challenger. When a_λ outputs β, b^2_λ outputs β as well if id^* was not queried to USKGen. Otherwise, b^2_λ outputs 0.

If $\mathbf{A}$ is generated as $\mathbf{A}^\top \xleftarrow{\$} \mathsf{OneSamp}(\lambda)$ (respectively, $\mathbf{A}^\top \xleftarrow{\$} \mathsf{ZeroSamp}(\lambda)$), the view of a_λ is the same as its view in G_3 (respectively, G_4). Hence, the probability that b^2_λ breaks the fine-grained matrix linear assumption is

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

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

Lemma 4.8 $\Pr[\mathsf{G}_5^{a_\lambda} \Rightarrow 1] = \Pr[\mathsf{G}_4^{a_\lambda} \Rightarrow 1]$.

Proof. In G_5, we do not use $(\mathbf{Y}_i)_{i=0}^{\ell}$ and $\mathbf{y}'$ in USKGen(id) or Enc(id^*) 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 K^* as in Figure 12. We now show that all the changes are purely conceptual.

In G_5, we generate $\mathbf{A}$ by sampling $\mathbf{R}_1 = \begin{pmatrix} \mathbf{I}_{\lambda-1} & 0 \\ \widetilde{\mathbf{r}}^\top & 1 \end{pmatrix}^\top \xleftarrow{\$} \mathsf{RSamp}(\lambda)$ and $\mathbf{R}_0 \xleftarrow{\$} \mathsf{LSamp}(\lambda)$, 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

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

and

$$\mathbf{c}_1^* = \sum_{i=1}^{\ell}\mathsf{id}_i^*\odot(\mathbf{Y}_i^\top \mid \mathbf{x}_i)(\mathbf{A}+\mathbf{N}^\lambda)\mathbf{r} = \sum_{i=1}^{\ell}\mathsf{id}_i^*\odot(\mathbf{Z}_i\mathbf{r}+\mathbf{x}_i).$$

Hence, the distributions of $\mathbf{Z}_i$ and $\mathbf{c}_1^*$ in G_5 remain the same, and the distributions of $\mathbf{z}'$ and K^* can be analyzed in the same way. The distribution of $\mathbf{v}$ does not change as well since

$$\begin{aligned}
\mathbf{v} &= \mathbf{t}^\top(\mathbf{Y}_0^\top + \sum_{i=1}^{\ell}\mathsf{id}_i\odot\mathbf{Y}_i^\top) + \mathbf{y}'^\top \\
&= \mathbf{t}^\top(\mathbf{Y}_0^\top + \mathbf{x}_0\cdot\widetilde{\mathbf{r}}^\top + \sum_{i=1}^{\ell}\mathsf{id}_i\odot(\mathbf{Y}_i^\top + \mathbf{x}_i\cdot\widetilde{\mathbf{r}}^\top)) + (\mathbf{y}'^\top + x'\cdot\widetilde{\mathbf{r}}^\top) \\
&- (\mathbf{t}^\top(\mathbf{x}_0 + \sum_{i=1}^{\ell}\mathsf{id}_i\odot\mathbf{x}_i) + x')\cdot\widetilde{\mathbf{r}}^\top \\
&= \mathbf{t}^\top(\mathbf{D}_0 + \sum_{i=1}^{\ell}\mathsf{id}_i\odot\mathbf{D}_i) + \mathbf{d}' - u\cdot\widetilde{\mathbf{r}}^\top.
\end{aligned}$$

Putting all above together, Lemma 4.8 immediately follows. □

Lemma 4.9 *There exists an adversary $\mathcal{B}_3 = \{b_\lambda^3\}_{\lambda\in\mathbb{N}} \in \mathsf{NC}^1$ such that b_λ^3 breaks the NC^1-(k,l)-PR-CMA-security of* MAC *with advantage*

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

Proof. The challenger of G_6 answers the $\textsc{Enc}(\mathsf{id}^*)$ query by choosing random K^*. We build b_λ^3 as in Figure 13 to show that the differences between G_6 and G_5 can be bounded by the advantage of breaking the PR-CMA security of MAC.

b_λ^3 runs in the same way as the challenger of 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_λ since $\mathbf{Y}_i$ and $\mathbf{y}'$ were uniformly sampled in G_5. Moreover, every time on receiving a query id to $\textsc{USKGen}$, b_λ^3 forwards id to its evaluation oracle $\textsc{Eval}$ to obtain the answer $(\mathbf{t}, u)$, and on receiving the query id^* to $\textsc{Enc}$, b_λ^3 forwards id^* to its challenge oracle $\textsc{Chal}$ and uses the answer $(\mathbf{h}_0, h_1)$ to simulate $\mathbf{r}$, $\mathbf{c}_1^*$, and K^* as in Figure 13. When a_λ outputs β, b_λ^3 outputs β as well if id^* was not queried to $\textsc{USKGen}$. Otherwise, b_λ^3 outputs 0.

If h_1 is uniform (i.e., b_λ^3 is in Game $\mathsf{PR\text{-}CMA}_{\mathsf{rand}}$) then the view of a_λ is identical to its view in G_6. If h_1 is real (i.e., b_λ^3 is in Game $\mathsf{PR\text{-}CMA}_{\mathsf{real}}$), then the view of a_λ is identical to its view in G_5. Thus the advantage of b_λ^3 is exactly

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

$\underline{\textsc{Init}}$:	$\underline{\textsc{Enc}(\mathsf{id}^*)}$: //one query
$\mathbf{R}_1 = \begin{pmatrix} \mathbf{I}_{\lambda-1} & \mathbf{0} \\ \widetilde{\mathbf{r}}^\top & 1 \end{pmatrix}^\top \xleftarrow{\$} \mathsf{RSamp}(\lambda)$,	$(\mathbf{h}_0, h_1) \xleftarrow{\$} \textsc{Chal}(\mathsf{id}^*)$
$\mathbf{R}_0 \xleftarrow{\$} \mathsf{LSamp}(\lambda), \mathbf{A}^\top = \mathbf{R}_0\mathbf{M}_0^\lambda\mathbf{R}_1$	$r_2, \cdots, r_n \xleftarrow{\$} \{0,1\}$
For $i = 0, \ldots, \ell$:	$\mathbf{r} = (1, r_2, \cdots, r_n)^\top$
$\mathbf{D}_i \xleftarrow{\$} \{0,1\}^{\lambda\times(\lambda-1)}$	$\mathbf{c}_0^* = (\mathbf{A} + \mathbf{N}^\lambda)\mathbf{r} \in \{0,1\}^\lambda$
$\mathbf{Z}_i = (\mathbf{0}\|\mathbf{D}_i)\mathbf{R}_0^\top \in \{0,1\}^{\lambda\times\lambda}$	$\mathbf{c}_1^* = \mathbf{Z}_0\mathbf{r} + \sum_{i=1}^{\ell} \mathsf{id}_i^* \odot \mathbf{Z}_i\mathbf{r} + \mathbf{h}_0^\top \in \{0,1\}^\lambda$
$\mathbf{d}' \xleftarrow{\$} \{0,1\}^{1\times(\lambda-1)}$	$\mathsf{K}^* = \mathbf{z}\cdot\mathbf{r} + h_1 \in \{0,1\}$
$\mathbf{z}' = (0\|\mathbf{d}')\mathbf{R}_0^\top \in \{0,1\}^{1\times\lambda}$	Return K^* and $\mathsf{ct}^* = (\mathbf{c}_0^*, \mathbf{c}_1^*)$
Return $\mathsf{pk} = (\mathbf{A}, (\mathbf{Z}_i)_{0\le i\le\ell}, \mathbf{z}')$	
$\underline{\textsc{USKGen}(\mathsf{id})}$:	$\underline{\textsc{Finalize}(\beta)}$:
$\mathcal{Q}_{\mathsf{id}} = \mathcal{Q}_{\mathsf{id}} \cup \{\mathsf{id}\}$	Return $(\mathsf{id}^* \notin \mathcal{Q}_{\mathsf{id}}) \wedge \beta$
$(\mathbf{t}, u) \xleftarrow{\$} \textsc{Eval}(\mathsf{id})$	
$\mathbf{v} = (\mathbf{D}_0 + \sum_{i=1}^{\ell} \mathsf{id}_i \odot \mathbf{D}_i)^\top\mathbf{t} + \mathbf{d}'^\top - \widetilde{\mathbf{r}}\cdot u \in \{0,1\}^{\lambda-1}$	
Return $\mathsf{usk}[\mathsf{id}] = (\mathbf{t}, u, \mathbf{v})$	

Figure 13: Description of $\mathcal{B}_3 = \{b_\lambda^3\}_{\lambda\in\mathbb{N}}$ (having access to the oracles $\textsc{Init}_{\mathsf{MAC}}$, $\textsc{Eval}$, $\textsc{Chal}$, $\textsc{Finalize}_{\mathsf{MAC}}$ of the $\mathsf{PR\text{-}CMA}_{\mathsf{real}}/\mathsf{PR\text{-}CMA}_{\mathsf{rand}}$ games of Figure 5 (instantiated with the encoding for equality predicate)) for the proof of Lemma 4.9.

Moreover, since all operations in b_λ^3 are performed in NC^1, we have $\mathcal{B}_3 = \{b_\lambda^3\}_{\lambda\in\mathbb{N}} \in \mathsf{NC}^1$, completing this part of proof. □

We now do all the previous steps in the reverse order as in Figure 14. Note that the view of the adversary in H_0 (respectively, H_4) is identical to its view in G_6 (respectively, $\mathsf{PR\text{-}ID\text{-}CPA}_{\mathsf{rand}}$). By using the above arguments in a reverse order, we have the following lemma.

Lemma 4.10 *There exists an adversary $\mathcal{B}_4 = \{b_\lambda^4\}_{\lambda\in\mathbb{N}} \in \mathsf{NC}^1$ such that b_λ^4 breaks the fine-grained matrix linear assumption with advantage*

$$(|\Pr[\mathsf{H}_4^{a_\lambda} \Rightarrow 1] - \Pr[\mathsf{H}_0^{a_\lambda} \Rightarrow 1]|)/2.$$

Putting all above together, Theorem 4.3 immediately follows. □

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 $\mathsf{AC}^0[2]$ 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 NC^1 with adaptive soundness. We refer the reader to Appendix B for details.

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 [36, 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 $\mathsf{AC}^0[2]$ and secure against NC^1 under $\mathsf{NC}^1 \subsetneq \oplus\mathsf{L/poly}$. We refer the reader to Appendix C 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 α-reconstruction properties hold in $GF(p)$ also hold in $GF(2)$ and by the standard linear-independence arguments in $GF(2)$, the α-privacy also holds in our case.

$\underline{\textsc{Init}:}$ //Games H_0, H_1-H_2, $\mathsf{H}_3 - \mathsf{H}_4$

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

$\mathsf{sk}_{\mathsf{MAC}} = (\mathbf{B}, \mathbf{x}_0, \ldots, \mathbf{x}_\ell, x') \xleftarrow{\$} \mathsf{Gen}_{\mathsf{MAC}\lambda}(\mathcal{G})$

For $i = 0, \ldots, \ell$:

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

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

$\mathsf{pk} = (\mathbf{A}, (\mathbf{Z}_i)_{0\le i\le \ell}, \mathbf{z}')$

$\mathsf{sk} = (\mathsf{sk}_{\mathsf{MAC}}, (\mathbf{Y}_i)_{0\le i\le \ell}, \mathbf{y}')$

Return pk

$\underline{\textsc{Finalize}(\beta):}$ //Games H_0-H_4

Return $(\mathsf{id}^* \notin \mathcal{Q}_{\mathsf{id}}) \wedge \beta$

$\underline{\textsc{USKGen}(\mathsf{id}):}$ //Games H_0-H_4

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

$\mathbf{v} = \mathbf{t}^\top(\mathbf{Y}_0^\top + \sum_{i=1}^{\ell} \mathsf{id}_i \odot \mathbf{Y}_i^\top) + \mathbf{y}'^\top \in \{0,1\}^{1\times(\lambda-1)}$

$\mathsf{usk}[\mathsf{id}] = (\mathbf{t}, u, \mathbf{v})$

Return $\mathsf{usk}[\mathsf{id}]$

$\underline{\textsc{Enc}(\mathsf{id}^*):}$ //Games H_0-H_1, H_2-H_3, H_4

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

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

$\mathbf{c}_1^* = (\mathbf{Y}_0^\top \mid \mathbf{x}_0)(\mathbf{A} + \mathbf{N}^\lambda)\mathbf{r} + \sum_{i=1}^{\ell} \mathsf{id}_i^* \odot (\mathbf{Y}_i^\top \mid \mathbf{x}_i)(\mathbf{A} + \mathbf{N}^\lambda)\mathbf{r} \in \{0,1\}^\lambda$

$\mathbf{c}_1^* = \mathbf{Z}_0\mathbf{r} + \sum_{i=1}^{\ell} \mathsf{id}_i^* \odot \mathbf{Z}_i\mathbf{r}$

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

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

Figure 14: Games H_0-H_4 for the proof of Theorem 4.3.

Let $\mathsf{PE} = \{\mathsf{rE}_\lambda, \mathsf{kE}_\lambda, \mathsf{sE}_\lambda, \mathsf{sD}_\lambda, \mathsf{rD}_\lambda\}_{\lambda\in\mathbb{N}} \in \mathsf{AC}^0[2]$ be a predicate encoding for $\mathsf{P} = \{\mathsf{p}_\lambda\}_{\lambda\in\mathbb{N}}$ with $\mathsf{rE}_\lambda : \mathcal{Y}\times\{0,1\}^\ell \to \{0,1\}^\eta$, $\mathsf{kE}_\lambda : \mathcal{Y}\times\{0,1\} \to \{0,1\}^\eta$, $\mathsf{sE}_\lambda : \mathcal{X}\times\{0,1\}^\ell \to \{0,1\}^\zeta$, $\mathsf{sD}_\lambda : \mathcal{X}\times\mathcal{Y}\times\{0,1\}^\zeta \to \{0,1\}$, and $\mathsf{rD}_\lambda : \mathcal{X}\times\mathcal{Y}\times\{0,1\}^\eta \to \{0,1\}$. Let $\mathsf{MAC}_{\mathsf{GA}} = \{\mathsf{Gen}_{\mathsf{MAC}\lambda}, \mathsf{Tag}_\lambda, \mathsf{Ver}_{\mathsf{MAC}\lambda}\}_{\lambda\in\mathbb{N}} \in \mathsf{AC}^0[2]$ be a PE-generalized affine MAC over $\{0,1\}^\lambda$ with message space $\mathcal{Y}$. Our ABKEM $\mathsf{ABKEM} = \{\mathsf{Gen}_\lambda, \mathsf{USKGen}_\lambda, \mathsf{Enc}_\lambda, \mathsf{Dec}_\lambda\}_{\lambda\in\mathbb{N}}$ is defined as in Figure 15.

$\underline{\mathsf{Gen}_\lambda:}$
$\mathbf{A}^\top \xleftarrow{\$} \mathsf{ZeroSamp}(\lambda)$
$\mathsf{sk}_{\mathsf{MAC}} = (\mathbf{B}, \mathbf{X}, x') \xleftarrow{\$} \mathsf{Gen}_{\mathsf{MAC}\lambda}(\mathsf{par})$
For $\mathbf{X} = (\mathbf{x}_1, \cdots, \mathbf{x}_\ell)$ and $i = 1, \ldots, \ell$:
  $\mathbf{Y}_i \xleftarrow{\$} \{0,1\}^{(\lambda-1)\times\lambda}$
  $\mathbf{Z}_i = (\mathbf{Y}_i^\top || \mathbf{x}_i)\mathbf{A} \in \{0,1\}^{\lambda\times\lambda}$
$\mathbf{y}' \xleftarrow{\$} \{0,1\}^{(\lambda-1)}$
$\mathbf{z}' = (\mathbf{y}'^\top || x')\mathbf{A} \in \{0,1\}^{1\times\lambda}$
$\mathsf{pk} = (\mathbf{A}, (\mathbf{Z}_i)_{1\le i\le\ell}, \mathbf{z}')$
$\mathsf{sk} = (\mathsf{sk}_{\mathsf{MAC}}, (\mathbf{Y}_i)_{1\le i\le\ell}, \mathbf{y}')$
Return $(\mathsf{pk}, \mathsf{sk})$

$\underline{\mathsf{USKGen}_\lambda(\mathsf{sk}, \mathsf{y} \in \mathcal{Y}):}$
$(\mathbf{t}, \mathbf{u}) \xleftarrow{\$} \mathsf{Tag}_\lambda(\mathsf{sk}_{\mathsf{MAC}}, \mathsf{y})$
$\mathbf{v} = \mathsf{rE}_\lambda(\mathsf{y}, \begin{pmatrix} \mathbf{t}^\top\mathbf{Y}_1^\top \\ \vdots \\ \mathbf{t}^\top\mathbf{Y}_\ell^\top \end{pmatrix})$
  $+ \mathsf{kE}_\lambda(\mathsf{y}, \mathbf{y}'^\top) \in \{0,1\}^{\eta\times(\lambda-1)}$
Return $\mathsf{usk}[\mathsf{y}] = (\mathbf{t}, \mathbf{u}, \mathbf{v})$

$\underline{\mathsf{Enc}_\lambda(\mathsf{pk}, \mathsf{x} \in \mathcal{X}):}$
$\mathbf{r} \xleftarrow{\$} \{0\} \times \{0,1\}^{\lambda-1}$
$\mathbf{c}_0 = \mathbf{A}\mathbf{r} \in \{0,1\}^\lambda$
$\mathbf{C}_1 = \mathsf{sE}_\lambda(\mathsf{x}, \begin{pmatrix} \mathbf{r}^\top\mathbf{Z}_1^\top \\ \vdots \\ \mathbf{r}^\top\mathbf{Z}_\ell^\top \end{pmatrix}) \in \{0,1\}^{\zeta\times\lambda}$
$\mathsf{K} = \mathbf{z}' \cdot \mathbf{r} \in \{0,1\}$.
Return K and $\mathsf{ct} = (\mathbf{c}_0, \mathbf{C}_1)$

$\underline{\mathsf{Dec}_\lambda(\mathsf{usk}[\mathsf{y}], \mathsf{y}, \mathsf{x}, \mathsf{ct}):}$
Parse $\mathsf{usk}[\mathsf{y}] = (\mathbf{t}, \mathbf{u}, \mathbf{v})$
Parse $\mathsf{ct} = (\mathbf{c}_0, \mathbf{C}_1)$
$\mathsf{K} = \mathsf{rD}_\lambda(\mathsf{x}, \mathsf{y}, \mathbf{v}||\mathbf{u})\mathbf{c}_0$
  $- \mathsf{sD}_\lambda(\mathsf{x}, \mathsf{y}, \mathbf{C}_1\mathbf{t}) \in \{0,1\}$
Return K

Figure 15: Construction of $\mathsf{ABKEM} = \{\mathsf{Gen}_\lambda, \mathsf{USKGen}_\lambda, \mathsf{Enc}_\lambda, \mathsf{Dec}_\lambda\}_{\lambda\in\mathbb{N}}$.

Theorem 5.1 *Under the assumption* $\mathsf{NC}^1 \subsetneq \oplus\mathsf{L/poly}$ *and the* NC^1-(k,l)-sE_λ-PR-CMA-*security of* $\mathsf{MAC}_{\mathsf{GA}}$*, where* k *is any constant and* $l = l(\lambda)$ *is any polynomial in* λ*,* ABKEM *is an* $\mathsf{AC}^0[2]$-*ABKEM that is* NC^1-(k,l)-PR-AT-CPA *secure against* NC^1*.*

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

By Equation (2) in Section 3.1, we have

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

and

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

Then, due to restricted α-reconstruction (see Definition 2.14), the difference of the above equations yields $\mathsf{K} = (\mathbf{y}'^\top||x')\mathbf{A}\mathbf{r} = \mathbf{z}' \cdot \mathbf{r}$, i.e., correctness is satisfied.

Let $\mathcal{A} = \{a_\lambda\}_{\lambda\in\mathbb{N}}$ be any adversary against the NC^1-(k,l)-PR-AT-CPA security of ABKEM. We now prove the NC^1-(k,l)-PR-AT-CPA security by defining a sequence of games G_0-G_6 as in Figure 16. Roughly, in the first four games, we show how to extract a challenge token for $\mathsf{MAC}_{\mathsf{GA}}$ 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_λ reveal no useful information other than the tags and token for MAC.

Lemma 5.2 $\Pr[\mathsf{PR\text{-}AT\text{-}CPA}^{a_\lambda}_{\mathsf{real}} \Rightarrow 1] = \Pr[\mathsf{G}_1^{a_\lambda} \Rightarrow 1] = \Pr[\mathsf{G}_0^{a_\lambda} \Rightarrow 1]$.

Proof. G_0 is the real attack game. In game G_1, we change the simulation of $\mathbf{c}_0^*$, $\mathbf{C}_1^*$ and K^* in $\textsc{Enc}(\mathsf{x})$ 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_λ in G_1 is identical to its view in G_0, completing this part of proof. □

Lemma 5.3 *There exists an adversary $\mathcal{B}_1 = \{b_\lambda^1\}_{\lambda\in\mathbb{N}} \in \mathsf{NC}^1$ such that b_λ^1 breaks the fine-grained matrix linear assumption (see Definition 2.5), which holds under $\mathsf{NC}^1 \subsetneq \oplus\mathsf{L}/\mathsf{poly}$ according to Theorem 2.13, with advantage*

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

Proof. G_1 and G_2 only differ in the distribution of $\mathbf{A}$, namely, $\mathbf{A}^\top \xleftarrow{\$} \mathsf{ZeroSamp}(\lambda)$ or $\mathbf{A}^\top \xleftarrow{\$} \mathsf{OneSamp}(\lambda)$, and we build the distinguisher b_λ^1 as follows.

b_λ^1 runs in exactly the same way as the challenger of G_1 except that in $\textsc{Init}$, instead of generating $\mathbf{A}$ by itself, it takes as input $\mathbf{A}^\top$ generated as $\mathbf{A}^\top \xleftarrow{\$} \mathsf{ZeroSamp}(\lambda)$ or $\mathbf{A}^\top \xleftarrow{\$} \mathsf{OneSamp}(\lambda)$ from its own challenger. When a_λ outputs β, b_λ^1 outputs β as well if no y such that $\mathsf{p}_\lambda(\mathsf{x},\mathsf{y}) = 1$ was queried to $\textsc{USKGen}$. Otherwise, b_λ^1 outputs 0.

If $\mathbf{A}$ is generated as $\mathbf{A}^\top \xleftarrow{\$} \mathsf{ZeroSamp}(\lambda)$ (respectively, $\mathbf{A}^\top \xleftarrow{\$} \mathsf{OneSamp}(\lambda)$), the view of a_λ is the same as its view in G_1 (respectively, G_2). Hence, the probability that b_λ^1 breaks the fine-grained matrix linear assumption is

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

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

Lemma 5.4 $\Pr[\mathsf{G}_3^{a_\lambda} \Rightarrow 1] = \Pr[\mathsf{G}_2^{a_\lambda} \Rightarrow 1]$.

Proof. In this game, we sample $\mathbf{r}$ in $\textsc{Enc}(\mathsf{x})$ as $\mathbf{r} \xleftarrow{\$} \{0,1\}^\lambda$ instead of $\mathbf{r} \xleftarrow{\$} \{0\} \times \{0,1\}^{\lambda-1}$. According to Lemma 2.9, the distributions of $\mathbf{A} + \mathbf{N}^\lambda$ in both G_2 and G_3 are identical to that of a matrix sampled from $\mathsf{ZeroSamp}$. Then this lemma follows from Lemma 2.11 immediately. □

Lemma 5.5 *There exists an adversary $\mathcal{B}_2 = \{b_\lambda^2\}_{\lambda\in\mathbb{N}} \in \mathsf{NC}^1$ such that b_λ^2 breaks the fine-grained matrix linear assumption with advantage*

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

Proof. G_1 and G_2 only differ in the distribution of $\mathbf{A}$, namely, $\mathbf{A}^\top \xleftarrow{\$} \mathsf{OneSamp}(\lambda)$ or $\mathbf{A}^\top \xleftarrow{\$} \mathsf{ZeroSamp}(\lambda)$, and we build the distinguisher b_λ^2 against Lemma 2.6 as follows.

b_λ^2 runs in exactly the same way as the challenger of G_3 except that in $\textsc{Init}$, instead of generating $\mathbf{A}$ by itself, it takes as input $\mathbf{A}^\top$ generated as $\mathbf{A}^\top \xleftarrow{\$} \mathsf{ZeroSamp}(\lambda)$ or $\mathbf{A}^\top \xleftarrow{\$} \mathsf{OneSamp}(\lambda)$ from its own challenger. When a_λ outputs β, b_λ^2 outputs β as well if no y such that $\mathsf{p}_\lambda(\mathsf{x},\mathsf{y}) = 1$ was queried to $\textsc{USKGen}$. Otherwise, b_λ^2 outputs 0.

If $\mathbf{A}$ is generated as $\mathbf{A}^\top \xleftarrow{\$} \mathsf{OneSamp}(\lambda)$ (respectively, $\mathbf{A}^\top \xleftarrow{\$} \mathsf{ZeroSamp}(\lambda)$), the view of a_λ is the same as its view in G_3 (respectively, G_4). Hence, the probability that b_λ^2 breaks the fine-grained matrix linear assumption is

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

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

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

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

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

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

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

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

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

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

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

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

Return pk

FINALIZE(β): //Games G_0-G_6

If $(\mathsf{p}_\lambda(\mathsf{x},\mathsf{y}) \neq 1$ for all $\mathsf{y} \in \mathcal{Q}_\mathsf{y}$, return β

Else return 0

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

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

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

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

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

Return $\mathsf{usk}[\mathsf{y}]$

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

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

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

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

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

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

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

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

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

Figure 16: Games G_0-G_6 for the proof of Theorem 5.1.

Lemma 5.6 $\Pr[\mathsf{G}_5^{a_\lambda} \Rightarrow 1] = \Pr[\mathsf{G}_4^{a_\lambda} \Rightarrow 1]$.

Proof. In G_5, we do not use $(\mathbf{Y}_i)_{i=1}^{\ell}$ and $\mathbf{y}'$ in USKGen(y) or Enc(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 K^* as in Figure 16. We now show that all the changes are purely conceptual.

In G_5, we generate $\mathbf{A}$ by sampling $\mathbf{R}_1 = \begin{pmatrix} \mathbf{I}_{\lambda-1} & 0 \\ \widetilde{\mathbf{r}}^\top & 1 \end{pmatrix}^\top \xleftarrow{\$} \mathsf{RSamp}(\lambda)$ and $\mathbf{R}_0 \xleftarrow{\$} \mathsf{LSamp}(\lambda)$, 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

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

and

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

Hence, the distributions of $\mathbf{Z}_i$ in G_5 remain the same, and the distributions of $\mathbf{z}'$ and K^* can be analyzed in the same way. The distribution of $\mathbf{v}$ does not change as well since

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

Putting all above together, Lemma 5.6 immediately follows. □

Lemma 5.7 *There exists an adversary* $\mathcal{B}_3 = \{b_\lambda^3\}_{\lambda \in \mathbb{N}} \in \mathsf{NC}^1$ *such that* b_λ^3 *breaks the* NC^1-(k,l)-PR-CMA *security of* $\mathsf{MAC}_{\mathsf{GA}}$ *with advantage*

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

Proof. The challenger of G_6 answers the Enc(x) query by choosing random K^*. We build b_λ^3 as in Figure 17 to show that the differences between G_6 and G_5 can be bounded by its advantage of breaking the PR-CMA security of $\mathsf{MAC}_{\mathsf{GA}}$.

b_λ^3 runs in the same way as the challenger of 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_λ since $\mathbf{Y}_i$ and $\mathbf{y}'$ were uniformly sampled in G_5. Moreover, every time on receiving a query y to USKGen, b_λ^3 forwards y to its evaluation oracle Eval to obtain the answer $(\mathbf{t}, \mathbf{u})$, and on receiving the query x to Enc, b_λ^3 forwards x to its challenge oracle Chal and uses the answer $(h, \mathbf{h}_0, h_1)$ to simulate $\mathbf{r}$, $\mathbf{C}_1^*$, and K^* as in Figure 17. When a_λ outputs β, b_λ^3 outputs β as well if no y such that $\mathsf{p}_\lambda(\mathsf{x}, \mathsf{y}) = 1$ was queried to USKGen. Otherwise, b_λ^3 outputs 0.

INIT:
$\mathbf{R}_1 = \begin{pmatrix} \mathbf{I}_{\lambda-1} & \mathbf{0} \\ \widetilde{\mathbf{r}}^\top & 1 \end{pmatrix}^\top \xleftarrow{\$} \mathsf{RSamp}(\lambda)$,
$\mathbf{R}_0 \xleftarrow{\$} \mathsf{LSamp}(\lambda), \mathbf{A}^\top = \mathbf{R}_0\mathbf{M}_0^\lambda\mathbf{R}_1$
For $i = 1, \ldots, \ell$:
$\mathbf{D}_i \xleftarrow{\$} \{0,1\}^{\lambda\times(\lambda-1)}$
$\mathbf{Z}_i = (\mathbf{0}\|\mathbf{D}_i)\mathbf{R}_0^\top \in \{0,1\}^{\lambda\times\lambda}$
$\mathbf{d}' \xleftarrow{\$} \{0,1\}^{1\times(\lambda-1)}, \mathbf{z}' = (0\|\mathbf{d}')\mathbf{R}_0^\top \in \{0,1\}^{1\times\lambda}$
$\mathsf{pk} = (\mathbf{A}, (\mathbf{Z}_i)_{1\le i\le\ell}, \mathbf{z}')$
Return pk
	ENC(x): //one query
	$(\mathbf{h}_0, h_1) \xleftarrow{\$}$ CHAL(x)
	$r_2, \cdots, r_n \xleftarrow{\$} \{0,1\}$
	$\mathbf{r} = (1, r_2, \cdots, r_n)^\top$
	$\mathbf{c}_0^* = (\mathbf{A} + \mathbf{N}^\lambda)\mathbf{r} \in \{0,1\}^\lambda$
	$\mathbf{C}_1^* = \mathsf{sE}_\lambda(\mathsf{x}, \begin{pmatrix} \mathbf{r}^\top\mathbf{Z}_1^\top \\ \vdots \\ \mathbf{r}^\top\mathbf{Z}_\ell^\top \end{pmatrix}) + \mathbf{h}_0 \in \{0,1\}^{\zeta\times\lambda}$
	$\mathsf{K}^* = \mathbf{z}' \cdot \mathbf{r} + h_1 \in \{0,1\}$
	Return K^* and $\mathsf{ct}^* = (\mathbf{c}_0^*, \mathbf{C}_1^*)$

USKGEN(y): $\mathcal{Q}_\mathsf{y} = \mathcal{Q}_\mathsf{y} \cup \{\mathsf{y}\}$ $(\mathbf{t}, \mathbf{u}) \xleftarrow{\$}$ EVAL(y) $\mathbf{v} = \mathsf{rE}_\lambda(\mathsf{y}, (\mathbf{D}_1^\top\mathbf{t}, \cdots, \mathbf{D}_\ell^\top\mathbf{t})^\top) + \mathsf{kE}_\lambda(\mathsf{y}, \mathbf{d}') - \mathbf{u}\cdot\widetilde{\mathbf{r}}^\top \in \{0,1\}^{\eta\times(\lambda-1)}$ $\mathsf{usk}[\mathsf{y}] = (\mathbf{t}, \mathbf{u}, \mathbf{v})$ Return $\mathsf{usk}[\mathsf{y}]$	FINALIZE(β): If $(\mathsf{p}_\lambda(\mathsf{x}, \mathsf{y}) \neq 1$ for all $\mathsf{y} \in \mathcal{Q}_\mathsf{y}$ return β Else return 0

Figure 17: Description of $\mathcal{B}_3 = \{b_\lambda^3\}_{\lambda\in\mathbb{N}}$ (having access to the oracles INIT_MAC, EVAL, CHAL, $\text{FINALIZE}_\mathsf{MAC}$ of the $\mathsf{PR\text{-}CMA}_\mathsf{real}$/$\mathsf{PR\text{-}CMA}_\mathsf{rand}$ games of Figure 5) for the proof of Lemma 5.7.

If h_1 is uniform (i.e., b_λ^3 is in Game $\mathsf{PR\text{-}CMA}_\mathsf{rand}$) then the view of a_λ is identical to its view in G_6. If h_1 is real (i.e., b_λ^3 is in Game $\mathsf{PR\text{-}CMA}_\mathsf{real}$) then the view of $\mathcal{A}$ is identical to its view in G_5. Hence, the advantage of b_λ^3 in breaking the $\mathsf{PR\text{-}CMA}$ security is

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

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

We now do all the previous steps in the reverse order as in Figure 18. Note that the view of the adversary in H_0 (respectively, H_4) is identical to its view in G_6 (respectively, $\mathsf{PR\text{-}AT\text{-}CPA}_\mathsf{rand}$). By using the above arguments in a reverse order, we have the following lemma.

Lemma 5.8 *There exists an adversary $\mathcal{B}_4 = \{b_\lambda^4\}_{\lambda\in\mathbb{N}} \in \mathsf{NC}^1$ such that b_λ^4 breaks the fine-grained matrix linear assumption with advantage*

$$(|\Pr[\mathsf{H}_4^{a_\lambda} \Rightarrow 1] - \Pr[\mathsf{H}_0^{a_\lambda} \Rightarrow 1]|)/2.$$

□

Putting all above together, Theorem 5.1 immediately follows. □

Acknowledgements

We would like to thank the anonymous reviewers for their valuable comments on a previous version of this paper. Parts of Yuyu Wang's work was supported by the National Natural Science Foundation for Young Scientists of China under Grant Number 62002049, the Natural Science Foundation of Sichuan under Grant Number 2023NSFSC0472, the Sichuan Science and Technology Program under Grant Number 2022YFG0037, and the National Key Research and Development Program of China under Grant Number 2022YFB3104600. Parts of Jiaxin Pan's work was supported by the Research Council of Norway under Project No. 324235. Parts of Yu Chen's work was supported by the National Key Research and Development Program of China under Grant Number 2021YFA1000600, the National Natural Science Foundation of China under Grant Number 62272269, Taishan Scholar Program of Shandong Province.

$\underline{\textsc{Init}:}$ //Games H_0, H_1-H_2, $\mathsf{H}_3 - \mathsf{H}_4$

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

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

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

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

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

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

Return pk

$\underline{\textsc{Finalize}(\beta):}$ //Games H_0-H_4

If $(\mathsf{p}_\lambda(\mathsf{x}, \mathsf{y}) \neq 1$ for all $\mathsf{y} \in \mathcal{Q}_\mathsf{y}$, return β

Else return 0

$\underline{\textsc{USKGen}(\mathsf{y}):}$ //Games H_0-H_4

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

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

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

Return $\mathsf{usk}[\mathsf{y}]$

$\underline{\textsc{Enc}(\mathsf{x}):}$ //Games H_0-H_1, H_2-H_3, H_4

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

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

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

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

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

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

Figure 18: Games H_0-H_4 for the proof of Theorem 5.1.

INIT:	//Games G_0, G_1-G_3, G_2-G_5, G_3-G_4, G_4-G_5, G_5
$\mathbf{M}^\top \xleftarrow{\$} \mathsf{ZeroSamp}(\lambda)$	$\mathbf{M}^\top \xleftarrow{\$} \mathsf{OneSamp}(\lambda)$ / $\mathbf{M}^\top \xleftarrow{\$} \mathsf{ZeroSamp}(\lambda)$
$\mathbf{w} \xleftarrow{\$} \{1\} \times \{0,1\}^{\lambda-1}$	$\mathbf{w} \xleftarrow{\$} \{0\} \times \{0,1\}^{\lambda-1}$ / $\mathbf{w} \xleftarrow{\$} \{1\} \times \{0,1\}^{\lambda-1}$
$\mathbf{x} \xleftarrow{\$} \mathbf{M}\mathbf{w}$	$\mathbf{x} \xleftarrow{\$} (\mathbf{M} + \mathbf{N}^\lambda)\mathbf{w}$
FINALIZE($\beta \in \{0,1\}$): Return β.	

Figure 19: Games G_0-G_5 for the proof of Proposition A.2.

Appendices

A The Proof of Theorem 2.13

Proof. We prove Theorem 2.13 by the following two propositions.

Proposition A.1 *For all* $\mathbf{M}^\top \in \mathsf{ZeroSamp}(\lambda)$ *and* $\mathbf{x} \in \mathsf{SampNo}_\lambda(\mathbf{M})$*, we have* $\mathbf{x} \in \{0,1\}^\lambda \setminus \mathrm{Im}(\mathbf{M})$*.*

of Proposition A.1. According to Lemma 2.10, we have $\mathrm{Im}(\mathbf{M}) = \{\mathbf{x} \mid \mathbf{w} \in \{0\} \times \{0,1\}^{\lambda-1}, \mathbf{x} = \mathbf{M}\mathbf{w}\}$. Since $\mathbf{N}^\lambda \mathbf{w} = \mathbf{0}$ for any $\mathbf{w} \in \{0\} \times \{0,1\}^{\lambda-1}$, we have $\mathrm{Im}(\mathbf{M}) = \{\mathbf{x} | \mathbf{w} \in \{0\} \times \{0,1\}^{\lambda-1}, \mathbf{x} = (\mathbf{M} + \mathbf{N}^\lambda)\mathbf{w}\}$. [3] Moreover, $(\mathbf{M} + \mathbf{N}^\lambda)$ is of full rank according to Lemma 2.9. Hence, for any $\mathbf{w} \in \{1\} \times \{0,1\}^{\lambda-1}$ and any $\mathbf{x} \in \mathrm{Im}(\mathbf{M})$, we have $(\mathbf{M} + \mathbf{N}^\lambda)\mathbf{w} \neq \mathbf{x}$. Namely, for any $\mathbf{w} \in \{1\} \times \{0,1\}^{\lambda-1}$, we have $(\mathbf{M} + \mathbf{N}^\lambda)\mathbf{w} \in \{0,1\}^\lambda \setminus \mathrm{Im}(\mathbf{M}^\top)$, completing the proof of Proposition A.1. □

Proposition A.2 *For any* $\mathcal{A} = \{a_\lambda\} \in \mathsf{NC}^1$*,*

$$|\Pr[a_\lambda(\mathbf{M}, \mathbf{x}_0) = 1] - \Pr[a_\lambda(\mathbf{M}, \mathbf{x}_1) = 1]| \leq \mathsf{negl}(\lambda)$$

where $\mathbf{M}^\top \xleftarrow{\$} \mathsf{ZeroSamp}(\lambda)$*,* $\mathbf{x}_0 \xleftarrow{\$} \mathsf{SampYes}_\lambda(\mathbf{M})$*, and* $\mathbf{x}_1 \xleftarrow{\$} \mathsf{SampNo}_\lambda(\mathbf{M})$*.*

of Proposition A.2. Let $\mathcal{A} = \{a_\lambda\}$ be any adversary in NC^1. We give intermediate games in Figure 19 to show that the advantage of $\mathcal{A}$ in breaking Proposition A.2 is negligible.

Lemma A.3 $\Pr[\mathsf{G}_1^{a_\lambda} \Rightarrow 1] = \Pr[\mathsf{G}_0^{a_\lambda} \Rightarrow 1]$.

Proof. In G_1 we sample $\mathbf{w} \xleftarrow{\$} \{0\} \times \{0,1\}^{\lambda-1}$ instead of $\mathbf{w} \xleftarrow{\$} \{1\} \times \{0,1\}^{\lambda-1}$. Then Lemma A.3 follows from the fact that the distributions of $\mathbf{x} = \mathbf{M}\mathbf{w}$ and $\mathbf{x}' = \mathbf{M}\mathbf{w}'$ are identical where $\mathbf{M}^\top \xleftarrow{\$} \mathsf{ZeroSamp}(\lambda)$, $\mathbf{w} \xleftarrow{\$} \{1\} \times \{0,1\}^{\lambda-1}$, and $\mathbf{w}' \xleftarrow{\$} \{0\} \times \{0,1\}^{\lambda-1}$, according to Lemma 2.11. □

Lemma A.4 $\Pr[\mathsf{G}_2^{a_\lambda} \Rightarrow 1] = \Pr[\mathsf{G}_1^{a_\lambda} \Rightarrow 1]$.

Proof. In G_2, we compute $\mathbf{x}$ as $\mathbf{x} = (\mathbf{M} + \mathbf{N}^\lambda)\mathbf{w}$ instead. Then Lemma A.4 follows from the fact that for any $\mathbf{w} \in \{0\} \times \{0,1\}^{\lambda-1}$, we have $\mathbf{N}^\lambda \mathbf{w} = \mathbf{0}$. □

Lemma A.5 *There exists an adversary* $\mathcal{B}_1 = \{b_\lambda^1\} \in \mathsf{NC}^1$ *such that* b_λ^1 *breaks Definition 2.5, which holds under* $\mathsf{NC}^1 \subsetneq \oplus\mathsf{L}/\mathsf{poly}$ *according to Lemma 2.6, with advantage*

$$|\Pr[\mathsf{G}_3^{a_\lambda} \Rightarrow 1] - \Pr[\mathsf{G}_2^{a_\lambda} \Rightarrow 1]|.$$

Proof. G_2 and G_3 only differ in the distribution of $\mathbf{M}$, namely, $\mathbf{M}^\top \xleftarrow{\$} \mathsf{ZeroSamp}(\lambda)$ or $\mathbf{M}^\top \xleftarrow{\$} \mathsf{OneSamp}(\lambda)$, and we build the distinguisher b_λ^1 as follows.

b_λ^1 runs in exactly the same way as the challenger of G_2 except that in INIT, instead of generating $\mathbf{M}$ by itself, it takes as input $\mathbf{M}^\top$ generated as $\mathbf{M}^\top \xleftarrow{\$} \mathsf{ZeroSamp}(\lambda)$ or $\mathbf{M}^\top \xleftarrow{\$} \mathsf{OneSamp}(\lambda)$ from its own challenger. When a_λ outputs β, b_λ^1 outputs β as well. If $\mathbf{M}$ is generated as $\mathbf{M}^\top \xleftarrow{\$} \mathsf{ZeroSamp}(\lambda)$

[3] See Section 2 for the notion of $\mathbf{N}^\lambda$.

(respectively, $\mathbf{M}^\top \xleftarrow{\$} \mathsf{OneSamp}(\lambda)$), the view of a_λ is the same as its view in G_2 (respectively, G_3). Hence, the probability that b^1_λ breaks the fine-grained matrix linear assumption is

$$|\Pr[\mathsf{G}_3^{a_\lambda} \Rightarrow 1] - \Pr[\mathsf{G}_2^{a_\lambda} \Rightarrow 1]|.$$

Moreover, since all operations in b^1_λ are performed in NC^1, we have $\mathcal{B}_1 = \{b^1_\lambda\}_{\lambda\in\mathbb{N}} \in \mathsf{NC}^1$, completing this part of proof. □

Lemma A.6 $\Pr[\mathsf{G}_4^{a_\lambda} \Rightarrow 1] = \Pr[\mathsf{G}_3^{a_\lambda} \Rightarrow 1]$.

Proof. In G_4 we sample $\mathbf{w} \xleftarrow{\$} \{1\} \times \{0,1\}^{\lambda-1}$ instead of $\mathbf{w} \xleftarrow{\$} \{0\} \times \{0,1\}^{\lambda-1}$.

Let $\mathbf{M}^\top \xleftarrow{\$} \mathsf{OneSamp}(\lambda)$. The distribution of $\mathbf{M} + \mathbf{N}^\lambda$ is identical to the output distribution of $\mathsf{ZeroSamp}(\lambda)$ according to Lemma 2.9. Therefore, according to Lemma 2.11, the distributions of $\mathbf{x} = (\mathbf{M} + \mathbf{N}^\lambda)\mathbf{w}$ and $\mathbf{x}' = (\mathbf{M} + \mathbf{N}^\lambda)\mathbf{w}'$ are identical for $\mathbf{w} \xleftarrow{\$} \{1\} \times \{0,1\}^{\lambda-1}$ and $\mathbf{w}' \xleftarrow{\$} \{0\} \times \{0,1\}^{\lambda-1}$, completing this part of proof. □

Lemma A.7 *There exists an adversary $\mathcal{B}_2 = \{b^2_\lambda\}_{\lambda\in\mathbb{N}}$ such that b^2_λ breaks the fine-grained matrix linear assumption with advantage*

$$|\Pr[\mathsf{G}_5^{a_\lambda} \Rightarrow 1] - \Pr[\mathsf{G}_4^{a_\lambda} \Rightarrow 1]|.$$

Proof. G_5 and G_4 only differ in the distribution of $\mathbf{M}$, namely, $\mathbf{M}^\top \xleftarrow{\$} \mathsf{OneSamp}(\lambda)$ or $\mathbf{M}^\top \xleftarrow{\$} \mathsf{ZeroSamp}(\lambda)$, and we build the distinguisher b^2_λ as follows.

b^2_λ runs in exactly the same way as the challenger of G_2 except that in Init, instead of generating $\mathbf{M}$ by itself, it takes as input $\mathbf{M}^\top$ generated as $\mathbf{M}^\top \xleftarrow{\$} \mathsf{OneSamp}(\lambda)$ or $\mathbf{M}^\top \xleftarrow{\$} \mathsf{ZeroSamp}(\lambda)$ from its own challenger. When a_λ outputs β, b^2_λ outputs β as well. If $\mathbf{M}$ is generated as $\mathbf{M}^\top \xleftarrow{\$} \mathsf{OneSamp}(\lambda)$ (respectively, $\mathbf{M}^\top \xleftarrow{\$} \mathsf{ZeroSamp}(\lambda)$), the view of a_λ is the same as its view in G_4 (respectively, G_5). Hence, the probability that b^2_λ breaks the fine-grained matrix linear assumption is

$$|\Pr[\mathsf{G}_5^{a_\lambda} \Rightarrow 1] - \Pr[\mathsf{G}_4^{a_\lambda} \Rightarrow 1]|.$$

Moreover, since all operations in b^2_λ are performed in NC^1, we have $\mathcal{B}_2 = \{b^2_\lambda\}_{\lambda\in\mathbb{N}} \in \mathsf{NC}^1$, completing this part of proof. □

Then Proposition A.2 follows from the fact that G_0 and G_5 are the real games of Proposition A.2, where the values $\mathbf{x}$ are sampled from $\mathsf{SampYes}_\lambda$ and SampNo_λ respectively. □

Putting all above together, Theorem 2.13 immediately follows. □

B Fine-grained Secure Quasi-Adaptive NIZK

In this section, we construct fine-grained QA-NIZK with adaptive soundness. We first give the definition of NC^1-QA-NIZK with adaptive soundness. Then we prove an NC^1 version of the Kernel Matrix Diffie-Hellman assumption [27], based on which we give a warm-up QA-NIZK in NC^1 with relatively low efficiency. Finally, we show how to achieve a more efficient construction.

B.1 Definitions

We now recall the definition of fine-grained QA-NIZK. Let $\mathcal{D}_\lambda$ be a probability distribution over a collection of relations $\mathsf{R} = \{\mathsf{R}_\mathbf{M}\}_{\mathbf{M}\in\mathcal{D}_\lambda}$ parametrized by a matrix $\mathbf{M} \in \{0,1\}^{n\times t}$ of rank $t' < n$ generated as $(\mathbf{M}^\top, \mathbf{M}^\perp) \xleftarrow{\$} \mathcal{D}_\lambda$ with the associated language

$$\mathcal{L}_\mathbf{M} = \{\mathbf{t} : \exists \mathbf{w} \in \{0,1\}^t, \text{ s.t. } \mathbf{t} = \mathbf{M}\mathbf{w}\}.$$

Witness sampleability. Notice that similar to witness sampleable distribution in the classical world [22], we require that $\mathcal{D}_\lambda$ additionally outputs a non-zero matrix $\mathbf{M}^\perp \in \{0,1\}^{n\times(n-t')}$ in the kernel of $\mathbf{M}^\top$. An example of sampleable distribution is $\mathsf{ZeroSamp}(n)$, which can additionally sample a non-zero vector in the kernel of its output. [4]

[4] In fact, the rightmost vector $(r_1, \cdots, r_{n-1}, 1)^\top$ of the intermediate matrix generated by $\mathsf{RSamp}(n)$ (see Figure 1) forms a vector in the kernel of $\mathbf{M}^\top$. See the proof of Lemma 3 in [15] for more details

$\textsc{Init}(\mathbf{M})$:	$\textsc{Finalize}(\mathbf{y}^*, \pi^*)$:
$(\mathsf{crs}, \mathsf{td}) \xleftarrow{\$} \mathsf{Gen}_\lambda(\mathbf{M})$ Return crs.	If $\mathbf{y}^* \notin \mathcal{L}_{\mathbf{M}}$ then return $\mathsf{Ver}_\lambda(\mathsf{crs}, \mathbf{y}^*, \pi^*)$ Else return 0

Figure 20: The AS security game for QANIZK.

Definition B.1 (Quasi-adaptive non-interactive zero-knowledge proof). A $\mathcal{C}_1$*-quasi-adaptive non-interactive zero-knowledge proof (QA-NIZK)* for a set of language distributions $\{\mathcal{D}_\lambda\}_{\lambda\in\mathbb{N}}$ is a function family $\mathsf{QANIZK} = \{\mathsf{Gen}_\lambda, \mathsf{Prove}_\lambda, \mathsf{Ver}_\lambda, \mathsf{Sim}_\lambda\}_{\lambda\in\mathbb{N}} \in \mathcal{C}_1$ with the following properties.

- $\mathsf{Gen}_\lambda(\mathbf{M})$ returns a CRS crs and a simulation trapdoor td.
- $\mathsf{Prove}_\lambda(\mathsf{crs},\mathbf{y}, \mathbf{w})$ returns a proof π.
- $\mathsf{Ver}_\lambda(\mathsf{crs},\mathbf{y}, \pi)$ deterministically returns 1 (accept) or 0 (reject).
- $\mathsf{Sim}_\lambda(\mathsf{crs}, \mathsf{td},\mathbf{y})$ returns a simulated proof π.

Perfect completeness is satisfied if for all $(\mathbf{M}^\top, \mathbf{M}^\perp) \in \mathcal{D}_\lambda$, all vectors $(\mathbf{y}, \mathbf{w})$ such that $\mathbf{y} = \mathbf{M}\mathbf{w}$, all $(\mathsf{crs}, \mathsf{td}) \in \mathsf{Gen}_\lambda(\mathbf{M})$, and all $\pi \in \mathsf{Prove}_\lambda(\mathsf{crs}, \mathbf{y}, \mathbf{w})$, we have

$$\mathsf{Ver}_\lambda(\mathsf{crs}, \mathbf{y}, \pi) = 1.$$

Perfect zero knowledge is satisfied if for all λ, all $(\mathbf{M}^\top, \mathbf{M}^\perp) \in \mathcal{D}_\lambda$, all $(\mathbf{y}, \mathbf{w})$ with $\mathbf{y} = \mathbf{M}\mathbf{w}$, and all $(\mathsf{crs}, \mathsf{td}) \in \mathsf{Gen}_\lambda(\mathbf{M})$, the following two distributions are identical:

$$\mathsf{Prove}_\lambda(\mathsf{crs}, \mathbf{y}, \mathbf{w}) \quad \text{and} \quad \mathsf{Sim}_\lambda(\mathsf{crs}, \mathsf{td}, \mathbf{y}).$$

Definition B.2 (Adaptive soundness for QANIZK). QANIZK is said to satisfy $\mathcal{C}_2$*-adaptive soundness* if for any adversary $\mathcal{A} = \{a_\lambda\}_{\lambda\in\mathbb{N}} \in \mathcal{C}_2$,

$$\Pr[\mathsf{AS}^{a_\lambda} \Rightarrow 1] \leq \mathsf{negl}(\lambda),$$

where Game AS is defined in Figure 20.

We note that in the above definition, the term "quasi-adaptive" means that the construction of the CRS depends on the statement $\mathbf{M}$. On the other hand, "adaptive" in the context of adaptive soundness means that in the soundness experiment, the adversary can choose the statement adaptively after seeing the CRS.

B.2 A Warm-up Construction

A new lemma. We now prove the following lemma under the assumption $\mathsf{NC}^1 \subsetneq \oplus\mathsf{L}/\mathsf{poly}$, based on which we can achieve adaptively sound QA-NIZKs in NC^1. It can be thought of as the counterpart of the Kernel Matrix Diffie-Hellman assumption [27] in NC^1.

Definition B.3 (Fine-grained kernel matrix assumption). If $\mathsf{NC}^1 \subsetneq \oplus\mathsf{L}/\mathsf{poly}$, then for all $\lambda \in \mathbb{N}$ and any adversary $\mathcal{A} = \{a_\lambda\}_{\lambda\in\mathbb{N}} \in \mathsf{NC}^1$, we have

$$\Pr[\mathbf{c}^\top\mathbf{M} = \mathbf{0} \wedge \mathbf{c} \neq \mathbf{0} | \mathbf{c} \xleftarrow{\$} a_\lambda(\mathbf{M})] \leq \mathsf{negl}(\lambda),$$

where $\mathbf{M}^\top \xleftarrow{\$} \mathsf{ZeroSamp}(\lambda)$.

Lemma B.4 *If* $\mathsf{NC}^1 \subsetneq \oplus\mathsf{L}/\mathsf{poly}$*, then the fine-grained kernel matrix assumption (see Definition B.3) holds.*

Proof. Let $\mathcal{A} = \{a_\lambda\}_{\lambda\in\mathbb{N}} \in \mathsf{NC}^1$ be an adversary such that a_λ breaks the fine-grained kernel matrix assumption with probability ϵ, we construct another adversary $\mathcal{B} = \{b_\lambda\}_{\lambda\in\mathbb{N}} \in \mathsf{NC}^1$ such that b_λ breaks the fine-grained subset membership problem (see Definition 2.12), which holds under $\mathsf{NC}^1 \subsetneq \oplus\mathsf{L}/\mathsf{poly}$ according to Theorem 2.13, with the same probability as follows.

On input $(\mathbf{M}, \mathbf{u})$ where $\mathbf{M}^\top \xleftarrow{\$} \mathsf{ZeroSamp}(\lambda)$ and $\mathbf{u} \xleftarrow{\$} \mathsf{SampYes}_\lambda(\mathbf{M})$ or $\mathbf{u} \xleftarrow{\$} \mathsf{SampNo}_\lambda(\mathbf{M})$, b_λ forwards $\mathbf{M}$ to a_λ. When a_λ outputs $\mathbf{c}$, b_λ outputs 1 iff the last element in $\mathbf{c}$ is 1, $\mathbf{c}^\top\mathbf{M} = \mathbf{0}$, and $\mathbf{c}^\top\mathbf{u} = 0$.

When $\mathbf{u} \xleftarrow{\$} \mathsf{SampYes}_\lambda(\mathbf{M})$, the probability that b_λ outputs 1 is ϵ. The reason is that when a_λ succeeds, we must have $\mathbf{c}^\top \mathbf{u} = 0$ when $\mathbf{c}^\top \mathbf{M} = \mathbf{0}$, and the last element of $\mathbf{c}$ must be 1 according to Lemma 2.7. Moreover, when $\mathbf{u} \xleftarrow{\$} \mathsf{SampNo}_\lambda(\mathbf{M})$, we have $\mathbf{u} = (\mathbf{M} + \mathbf{N}^\lambda)\mathbf{w}$ for some $\mathbf{w} \in \{1\} \times \{0,1\}^{\lambda-1}$. If $\mathbf{c}^\top \mathbf{M} = \mathbf{0}$, we have $\mathbf{c}^\top \mathbf{u} = \mathbf{c}^\top \mathbf{N}^\lambda \mathbf{w} = \mathbf{c}^\top (0, \cdots, 0, 1)^\top$, i.e., either $\mathbf{c}^\top \mathbf{u} = 1$ or the last element of $\mathbf{c}$ is 0. Hence, b_λ outputs 0 anyway when $\mathbf{u} \xleftarrow{\$} \mathsf{SampNo}_\lambda(\mathbf{M})$. Therefore, we have

$$|\Pr[b_\lambda(\mathbf{x}) = 1 \mid \mathbf{x} \xleftarrow{\$} \mathsf{SampYes}_\lambda(\lambda)] - \Pr[b_\lambda(\mathbf{x}) = 1 \mid \mathbf{x} \xleftarrow{\$} \mathsf{SampNo}_\lambda(\lambda)]| = \epsilon.$$

Moreover, since all operations in b_λ are performed in NC^1, we have $\mathcal{B} = \{b_\lambda\}_{\lambda \in \mathbb{N}} \in \mathsf{NC}^1$, completing the proof of Lemma B.4. □

Constructing QA-NIZK based on Lemma B.4. Based on the above lemma, we can easily achieve NC^1-QA-NIZKs with adaptive soundness, one-time simulation soundness, and unbounded simulation soundness against NC^1 by adopting the techniques in [23]. [5] Specifically, we only have to move the algorithms in [23] from $GF(p)$ for a large prime p to $GF(2)$, change the matrix Diffie-Hellman distributions to $\mathsf{SampYes}_\lambda(\lambda)$, and generate a large number of proofs in parallel to bound the advantage of the adversary. We now give an adaptively sound QA-NIZK QANIZK_0 w.r.t. a set of (sampleable) distributions $\{\mathcal{D}_\lambda\}_{\lambda \in \mathbb{N}}$ in Figure 21 as an instance. [6]

$\mathsf{Gen}_\lambda(\mathbf{M} \in \{0,1\}^{n \times t})$:	$\mathsf{Prove}_\lambda(\mathsf{crs}, \mathbf{y} \in \{0,1\}^n, \mathbf{x} \in \{0,1\}^t)$:
$\mathbf{A}^\top \xleftarrow{\$} \mathsf{ZeroSamp}(\lambda)$	For $i = 1, \cdots, \ell$
For $i = 1, \cdots, \ell$	$\pi_i = \mathbf{x}^\top \mathbf{P}_i \in \{0,1\}^{1 \times \lambda}$
$\mathbf{K}_i \xleftarrow{\$} \{0,1\}^{n \times \lambda}$	Return $\pi = (\pi_i)_{i=1}^\ell$
$\mathbf{P}_i = \mathbf{M}^\top \mathbf{K}_i \in \{0,1\}^{t \times \lambda}$	
$\mathbf{C}_i = \mathbf{K}_i \mathbf{A} \in \{0,1\}^{n \times \lambda}$	$\mathsf{Sim}_\lambda(\mathsf{crs}, \mathsf{td}, \mathbf{y})$:
Return $\mathsf{crs} = (\mathbf{A}, (\mathbf{P}_i, \mathbf{C}_i)_{i=1}^\ell)$ and $\mathsf{td} = (\mathbf{K}_i)_{i=1}^\ell$	For $i = 1, \cdots, \ell$
	$\pi_i = \mathbf{y}^\top \mathbf{K}_i \in \{0,1\}^{1 \times \lambda}$
	Return $\pi = (\pi_i)_{i=1}^\ell$
	$\mathsf{Ver}_\lambda(\mathsf{crs}, \mathbf{y}, \pi)$:
	If $\pi_i \mathbf{A} = \mathbf{y}^\top \mathbf{C}_i$ for $i = 1, \cdots, \ell$
	return 1
	Else return 0

Figure 21: Definition of $\mathsf{QANIZK}_0 = \{\mathsf{Gen}_\lambda, \mathsf{Prove}_\lambda, \mathsf{Ver}_\lambda, \mathsf{Sim}_\lambda\}_{\lambda \in \mathbb{N}}$. We require that $2^{\ell(\cdot)}$ is some super-polynomial in λ.

Theorem B.5 *If $\mathsf{NC}^1 \subsetneq \oplus\mathsf{L/poly}$, then QANIZK_0 is an $\mathsf{AC}^0[2]$-QA-NIZK that is NC^1-adaptively sound for all $\mathbf{M}$ in the distributions $\{\mathcal{D}_\lambda\}_{\lambda \in \mathbb{N}}$ (see Appendix B.1 for the definition of $\{\mathcal{D}_\lambda\}_{\lambda \in \mathbb{N}}$).*

Proof. First, we note that $\{\mathsf{Gen}_\lambda\}_{\lambda \in \mathbb{N}}$, $\{\mathsf{Prove}_\lambda\}_{\lambda \in \mathbb{N}}$, $\{\mathsf{Sim}_\lambda\}_{\lambda \in \mathbb{N}}$, and $\{\mathsf{Ver}_\lambda\}_{\lambda \in \mathbb{N}}$ are computable in $\mathsf{AC}^0[2]$, since they only involve operations including multiplication of a constant number of matrices and sampling random bits.

Perfect correctness and perfect zero-knowledge follow from the fact that for all $\mathbf{y} = \mathbf{M}\mathbf{x}$ and $\mathbf{P}_i = \mathbf{M}^\top \mathbf{K}_i$, we have

$$\mathbf{x}^\top \mathbf{P}_i = \mathbf{x}^\top \mathbf{M}^\top \mathbf{K}_i = \mathbf{y}^\top \mathbf{K}_i.$$

Let $\mathcal{A} = \{a_\lambda\}_{\lambda \in \mathbb{N}}$ be an adversary breaking the adaptive soundness of QANIZK_0 with advantage ϵ, we have the following lemma.

Lemma B.6 *There exists an adversary $\mathcal{B} = \{b_\lambda\}_{\lambda \in \mathbb{N}} \in \mathsf{NC}^1$ such that b_λ breaks the fine-grained kernel matrix assumption (see Definition B.3), which holds under $\mathsf{NC}^1 \subsetneq \oplus\mathsf{L/poly}$ according to Lemma B.4, with probability $\epsilon - 1/2^\ell$.*

[5] One-time (respectively, unbounded) simulation soundness prevents the adversary from proving a false statement after seeing a single simulated proof for a statement (respectively, multiple simulated proofs for statements) of its choice. We refer the reader to [23] for the formal definitions.

[6] We do not exploit the sampleability of the distribution for this construction.

Proof. We construct b_λ as follows.

b_λ on input $\mathbf{A}$ samples $(\mathbf{M}^\top, \mathbf{M}^\perp) \xleftarrow{\$} \mathcal{D}_\lambda$ and $\mathbf{K}_i \xleftarrow{\$} \{0,1\}^{n\times\lambda}$, and sets $\mathbf{P}_i = \mathbf{M}^\top\mathbf{K}_i$ and $\mathbf{C}_i = \mathbf{K}_i\mathbf{A}$ for all $i \in [\ell]$. Then it sends $\mathsf{crs} = (\mathbf{A}, (\mathbf{P}_i, \mathbf{C}_i)_{i=1}^\ell)$ to a_λ. When a_λ outputs $(\pi = (\pi_i)_{i=1}^\ell, \mathbf{y})$, b_λ searches j such that

$$\pi_j\mathbf{A} = \mathbf{y}^\top\mathbf{C}_j = \mathbf{y}^\top\mathbf{K}_j\mathbf{A}$$

and

$$\pi_j - \mathbf{y}^\top\mathbf{K}_j \neq \mathbf{0}.$$

If the searching procedure fails, b_λ aborts. b_λ then outputs $\pi_j - \mathbf{y}^\top\mathbf{K}_j$.

When a_λ succeeds, we have $\pi_j\mathbf{A} = \mathbf{y}^\top\mathbf{C}_j$ for all j and $\mathbf{y} \notin \mathrm{Im}(\mathbf{M})$. Let $\hat{\mathbf{a}}$ be a fixed non-zero vector such that $\hat{\mathbf{a}} \notin \mathrm{Im}(\mathbf{A})$. For each i, since a_λ learns no information on $\mathbf{K}_i$ other than $\mathbf{M}^\top\mathbf{K}_i$ and $\mathbf{K}_i\mathbf{A}$, $\mathbf{y}^\top\mathbf{K}_i\hat{\mathbf{a}}$ is information-theoretically hidden in the view of a_λ, i.e., the probability that there exists j such that $\pi_j\hat{\mathbf{a}} - \mathbf{y}^\top\mathbf{K}_j\hat{\mathbf{a}} \neq 0$ is at least $1/2^\ell$. Since $\pi_j\hat{\mathbf{a}} - \mathbf{y}^\top\mathbf{K}_j\hat{\mathbf{a}} \neq 0$ implies $\pi_j - \mathbf{y}^\top\mathbf{K}_j \neq \mathbf{0}$, the probability that b_λ breaks the fine-grained kernel matrix assumption is at least $\epsilon - 1/2^\ell$, completing this part of proof. □

Since the fine-grained kernel matrix assumption holds if $\mathsf{NC}^1 \subsetneq \oplus\mathsf{L}/\mathsf{poly}$ according to Lemma B.4, putting all above together, Theorem B.5 immediately follows. □

B.3 A More Efficient Construction

A disadvantage of the scheme in Appendix B.2 is that we have to generate a large number of proofs in parallel. In this section, we give a more efficient NC^1-adaptively sound QA-NIZK $\mathsf{QANIZK}_1 = \{\mathsf{Gen}_\lambda, \mathsf{Prove}_\lambda, \mathsf{Ver}_\lambda, \mathsf{Sim}_\lambda\}_{\lambda\in\mathbb{N}}$ w.r.t. a set of distributions $\{\mathcal{D}_\lambda\}_{\lambda\in\mathbb{N}}$ in Figure 22. As in Definition B.1, we require that $\mathcal{D}_\lambda$ be witness sampleable, i.e., it outputs a matrix $\mathbf{M} \in \{0,1\}^{n\times t}$ of rank $t' < n$ additionally with a matrix (or vector) $\mathbf{M}^\perp \in \{0,1\}^{n\times(n-t')}$ with rank $n - t'$ in its kernel.

The proof size of this construction is $(\lambda - 1)\cdot(n - t')$. Since $\mathbf{M}$ (or $\mathbf{M}^\top$) is usually a combination of matrices sampled from $\mathsf{ZeroSamp}(\lambda)$ in NC^1, $n - t'$ is typically a constant number. For instance, when proving that two ciphertexts of the PKE scheme in [13] correspond to the same message or proving the validity of a public key of the PKE scheme in [15], the proof size is only $\lambda - 1$ in contrast to $\lambda\cdot\ell$ for a large number ℓ in the warm-up construction.

$\underline{\mathsf{Gen}_\lambda(\mathbf{M} \in \{0,1\}^{n\times t})}$: $\mathbf{A}^\top \xleftarrow{\$} \mathsf{ZeroSamp}(\lambda)$ For $i = 1, \cdots, n - t'$ $\quad\mathbf{K}_i \xleftarrow{\$} \{0,1\}^{n\times(\lambda-1)}$ $\quad\mathbf{P}_i = \mathbf{M}^\top\mathbf{K}_i \in \{0,1\}^{t\times(\lambda-1)}$ $\quad\mathbf{C}_i = (\mathbf{K}_i\|\mathbf{0})\mathbf{A} \in \{0,1\}^{n\times(\lambda-1)}$ Return $\mathsf{crs} = (\mathbf{A}, (\mathbf{P}_i, \mathbf{C}_i)_{i=1}^{n-t'})$ and $\mathsf{td} = (\mathbf{K}_i)_{i=1}^{n-t'}$	$\underline{\mathsf{Prove}_\lambda(\mathsf{crs}, \mathbf{y} \in \{0,1\}^n, \mathbf{x} \in \{0,1\}^t)}$: For $i = 1, \cdots, n - t'$ $\quad\pi_i = \mathbf{x}^\top\mathbf{P}_i \in \{0,1\}^{1\times(\lambda-1)}$ Return $\pi = (\pi_i)_{i=1}^{n-t'}$ $\underline{\mathsf{Sim}_\lambda(\mathsf{crs}, \mathsf{td}, \mathbf{y})}$: For $i = 1, \cdots, n - t'$ $\quad\pi_i = \mathbf{y}^\top\mathbf{K}_i \in \{0,1\}^{1\times(\lambda-1)}$ Return $\pi = (\pi_i)_{i=1}^{n-t'}$ $\underline{\mathsf{Ver}_\lambda(\mathsf{crs}, \mathbf{y}, \pi)}$: If $(\pi_i\|0)\mathbf{A} = \mathbf{y}^\top\mathbf{C}_i$ for $i = 1, \cdots, n - t'$, return 1 Else return 0

Figure 22: Definition of $\mathsf{QANIZK}_1 = \{\mathsf{Gen}_\lambda, \mathsf{Prove}_\lambda, \mathsf{Ver}_\lambda, \mathsf{Sim}_\lambda\}_{\lambda\in\mathbb{N}}$.

Theorem B.7 *If* $\mathsf{NC}^1 \subsetneq \oplus\mathsf{L}/\mathsf{poly}$*, then* QANIZK_1 *is an* $\mathsf{AC}^0[2]$*-QA-NIZK that is* NC^1*-adaptively sound for all* $\mathbf{M}$ *in the distributions* $\{\mathcal{D}_\lambda\}_{\lambda\in\mathbb{N}}$ *(see Appendix B.1 for the definition of* $\{\mathcal{D}_\lambda\}_{\lambda\in\mathbb{N}}$*).*

Proof. First, we note that $\{\mathsf{Gen}_\lambda\}_{\lambda\in\mathbb{N}}$, $\{\mathsf{Prove}_\lambda\}_{\lambda\in\mathbb{N}}$, $\{\mathsf{Sim}_\lambda\}_{\lambda\in\mathbb{N}}$, and $\{\mathsf{Ver}_\lambda\}_{\lambda\in\mathbb{N}}$ are computable in $\mathsf{AC}^0[2]$, since they only involve operations including multiplications of a constant number of matrices and sampling random bits.

Perfect correctness and perfect zero-knowledge follow from the fact that for all $\mathbf{y} = \mathbf{M}\mathbf{x}$ and $\mathbf{P}_i = \mathbf{M}^\top \mathbf{K}_i$, we have

$$\mathbf{x}^\top \mathbf{P}_i = \mathbf{x}^\top \mathbf{M}^\top \mathbf{K}_i = \mathbf{y}^\top \mathbf{K}_i.$$

Let $\mathcal{A} = \{a_\lambda\}_{\lambda \in \mathbb{N}} \in \mathsf{NC}^1$ be any adversary against the NC^1-adaptive soundness of QANIZK_1. We now show that QANIZK_1 is adaptively sound against NC^1 via a sequence of hybrid games as in Figure 23. The crucial step is to use the technique exploited by our IBKEM to switch $(\mathbf{K}_i||\mathbf{0})\mathbf{A}$ to $(\mathbf{0}||\mathbf{K}_i)\mathbf{R}_0^\top$, and then switch it back to $(\mathbf{K}'_i||\mathbf{M}^\perp \mathbf{e}_i)\mathbf{A}$ for $\mathbf{K}_i = \mathbf{K}'_i + \mathbf{M}^\perp \mathbf{e}_i \cdot \widetilde{\mathbf{r}}^\top$, where $\mathbf{R}_0$ and $\widetilde{\mathbf{r}}$ are intermediate values generated during the sampling procedure for $\mathbf{A}$.

<u>INIT:</u> // Games G_0, $\boxed{\mathsf{G}_1\text{-}\mathsf{G}_2}$, $\boxed{\mathsf{G}_2\text{-}\mathsf{G}_3}$, G_3

$\mathbf{A}^\top \xleftarrow{\$} \mathsf{ZeroSamp}(\lambda)$

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

For $i = 1, \cdots, n - t'$

$\mathbf{K}_i \xleftarrow{\$} \{0,1\}^{n \times (\lambda-1)}$

$\boxed{\mathbf{K}'_i \xleftarrow{\$} \{0,1\}^{n \times (\lambda-1)}, \mathbf{K}_i = \mathbf{K}'_i + \mathbf{M}^\perp \mathbf{e}_i \cdot \widetilde{\mathbf{r}}^\top}$

$\mathbf{P}_i = \mathbf{M}^\top \mathbf{K}_i \in \{0,1\}^{t \times (\lambda-1)}$ $\quad \mathbf{P}_i = \mathbf{M}^\top \mathbf{K}'_i$

$\mathbf{C}_i = (\mathbf{K}_i||\mathbf{0})\mathbf{A} \in \{0,1\}^{n \times \lambda}$ $\quad \boxed{\mathbf{C}_i = (\mathbf{0}||\mathbf{K}_i)\mathbf{R}_0^\top}$ $\quad \mathbf{C}_i = (\mathbf{K}'_i||\mathbf{M}^\perp \mathbf{e}_i)\mathbf{A}$

Return $\mathsf{crs} = (\mathbf{A}, (\mathbf{P}_i, \mathbf{C}_i)_{i=1}^{n-t'})$, $\mathsf{td} = (\mathbf{K}_i)_{i=1}^{n-t'}$

<u>FINALIZE$(\mathbf{z}, \mathbf{y})$:</u> // Games G_0-G_2

For $i = 1, \cdots, n - t'$

If $(\pi_i||0)\mathbf{A} = \mathbf{y}^\top \mathbf{C}_i$

return 1

Figure 23: Games $\mathsf{G}_0, \mathsf{G}_1, \mathsf{G}_2$ for the proof of Theorem B.7. $\mathbf{e}_i \in \{0,1\}^{n-t'}$ denotes the vector with the ith element being 1 and the others being 0.

Lemma B.8 $\Pr[\mathsf{AS}^{a_\lambda} \Rightarrow 1] = \Pr[\mathsf{G}_1^{a_\lambda} \Rightarrow 1] = \Pr[\mathsf{G}_0^{a_\lambda} \Rightarrow 1]$.

Proof. In G_1, we generate $\mathbf{A}$ by sampling $\mathbf{R}_1 = \begin{pmatrix} \mathbf{I}_{\lambda-1} & 0 \\ \widetilde{\mathbf{r}}^\top & 1 \end{pmatrix}^\top \xleftarrow{\$} \mathsf{RSamp}(\lambda)$ and $\mathbf{R}_0 \xleftarrow{\$} \mathsf{LSamp}(\lambda)$, and setting $\mathbf{A}^\top = \mathbf{R}_0 \mathbf{M}_0^\lambda \mathbf{R}_1$. Moreover, for all i, we replace $\mathbf{C}_i = (\mathbf{K}_i||\mathbf{0})\mathbf{A}$ by $\mathbf{C}_i = (\mathbf{0}||\mathbf{K}_i)\mathbf{R}_0^\top$.

The view of $\mathcal{A}$ in this game is identical to its view in G_0 since the way we generate $\mathbf{A}$ is exactly the "zero-sampling" procedure, and we have

$$\begin{aligned}
\mathbf{C}_i &= (\mathbf{K}_i||\mathbf{0})\mathbf{A} = (\mathbf{K}_i||\mathbf{0})\mathbf{R}_1^\top {\mathbf{M}_0^\lambda}^\top \mathbf{R}_0^\top \\
&= (\mathbf{K}_i||\mathbf{0}) \begin{pmatrix} \mathbf{I}_{\lambda-1} & \mathbf{0} \\ \widetilde{\mathbf{r}}^\top & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \\ 0 & \cdots & 0 & & 1 \\ 0 & & \cdots & & 0 \end{pmatrix} \mathbf{R}_0^\top \\
&= (\mathbf{K}_i||\mathbf{0}) \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \\ 0 & \cdots & 0 & & 1 \\ 0 & & \cdots & & 0 \end{pmatrix} \mathbf{R}_0^\top \\
&= (\mathbf{0}||\mathbf{K}_i)\mathbf{R}_0^\top.
\end{aligned}$$

□

Lemma B.9 $\Pr[\mathsf{G}_2^{a_\lambda} \Rightarrow 1] = \Pr[\mathsf{G}_1^{a_\lambda} \Rightarrow 1]$.

Proof. In G_2, for all i, instead of generating $\mathbf{K}_i$ as a uniformly random matrix, we generate $\mathbf{K}_i$ by randomly sampling $\mathbf{K}'_i \xleftarrow{\$} \{0,1\}^{n\times(\lambda-1)}$ and setting $\mathbf{K}_i = \mathbf{K}'_i + \mathbf{M}^\perp \mathbf{e}_i \cdot \widetilde{\mathbf{r}}^\top$, where $\mathbf{e}_i \in \{0,1\}^{n-t'}$ denotes the vector with the ith element being 1 and the other bits being 0. Since the distribution of $\mathbf{K}_i$ is still uniform, the view of $\mathcal{A}$ remains the same. □

Lemma B.10 $\Pr[\mathsf{G}_3^{a_\lambda} \Rightarrow 1] = \Pr[\mathsf{G}_2^{a_\lambda} \Rightarrow 1]$.

Proof. This lemma follows from the fact that for all i, we have

$$\mathbf{M}\mathbf{K}_i = \mathbf{M}(\mathbf{K}'_i + \mathbf{M}^\perp \mathbf{e}_i \cdot \widetilde{\mathbf{r}}^\top) = \mathbf{M}\mathbf{K}'_i$$

and

$$\begin{aligned}
\mathbf{C}_i &= (\mathbf{0}||\mathbf{K}'_i + \mathbf{M}^\perp \mathbf{e}_i \cdot \widetilde{\mathbf{r}}^\top)\mathbf{R}_0^\top \\
&= (\mathbf{K}'_i + \mathbf{M}^\perp \mathbf{e}_i \cdot \widetilde{\mathbf{r}}^\top||\mathbf{M}^\perp \mathbf{e}_i) \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \\ 0 & \cdots & 0 & & 1 \\ 0 & & \cdots & & 0 \end{pmatrix} \mathbf{R}_0^\top \\
&= (\mathbf{K}'_i||\mathbf{M}^\perp \mathbf{e}_i) \begin{pmatrix} \mathbf{I}_{\lambda-1} & \mathbf{0} \\ \widetilde{\mathbf{r}}^\top & 1 \end{pmatrix} \mathbf{M}_0^{\lambda\top} \mathbf{R}_0^\top \\
&= (\mathbf{K}'_i||\mathbf{M}^\perp \mathbf{e}_i)\mathbf{A}
\end{aligned}$$

□

Lemma B.11 *There exists an adversary $\mathcal{B} = \{b_\lambda\}_{\lambda\in\mathbb{N}} \in \mathsf{NC}^1$ such that b_λ breaks the fine-grained kernel matrix assumption (see Definition B.3), which holds under $\mathsf{NC}^1 \subsetneq \oplus\mathsf{L}/\mathsf{poly}$ according to Lemma B.4, with probability* $\Pr[\mathsf{G}_3^{a_\lambda} \Rightarrow 1]$.

Proof. We construct b_λ as follows.

b_λ on input $\mathbf{A}$ samples $(\mathbf{M}^\top, \mathbf{M}^\perp) \xleftarrow{\$} \mathcal{D}_\lambda$ and $\mathbf{K}'_i \xleftarrow{\$} \{0,1\}^{n\times(\lambda-1)}$, and sets $\mathbf{P}_i = \mathbf{M}^\top(\mathbf{K}'_i||\mathbf{0})$ and $\mathbf{C}_i = (\mathbf{K}'_i||\mathbf{M}^\perp \mathbf{e}_i)\mathbf{A}$ for all i. Then it sends $\mathsf{crs} = (\mathbf{A}, (\mathbf{P}_i, \mathbf{C}_i)_{i=1}^{n-t'})$ to a_λ. When a_λ outputs $(\pi = (\pi_i)_{i=1}^{n-t'}, \mathbf{y})$, b_λ searches j such that

$$(\pi_j||0)\mathbf{A} = \mathbf{y}^\top \mathbf{C}_j = \mathbf{y}^\top(\mathbf{K}'_j||\mathbf{M}^\perp \mathbf{e}_j)\mathbf{A}$$

and

$$\pi_j||0 - \mathbf{y}^\top(\mathbf{K}'_j||\mathbf{M}^\perp \mathbf{e}_j) \neq \mathbf{0}.$$

If the searching procedure fails, b_λ aborts. b then outputs $\pi_j||0 - \mathbf{y}^\top(\mathbf{K}'_j||\mathbf{M}^\perp \mathbf{e}_j)$.

Since all the operations performed by b_λ are in NC^1, we have $\mathcal{B} \in \mathsf{NC}^1$.

When a_λ succeeds, we have $(\pi_j||0)\mathbf{A} = \mathbf{y}^\top \mathbf{C}_j$ for all j and $\mathbf{y} \notin \mathsf{Span}(\mathbf{M})$. In this case, $\mathbf{y}^\top \mathbf{M}^\perp \neq \mathbf{0}$, i.e., there must exists j such that $\mathbf{y}^\top \mathbf{M}^\perp \mathbf{e}_i = 1$. Hence the probability that b_λ breaks the fine-grained kernel matrix assumption is exactly $\Pr[\mathsf{G}_3^{a_\lambda} \Rightarrow 1]$, completing this part of proof. □

Putting all above together, Theorem B.7 immediately follows. □

Concurrent fine-grained NIZKs. Assuming $\mathsf{NC}^1 \subsetneq \oplus\mathsf{L}/\mathsf{poly}$, our work presents an efficient QA-NIZK that achieves perfect zero-knowledge and can handle languages expressible as linear subspaces. Below we compare our QA-NIZK to other existing fine-grained NIZKs [2, 32, 33].

Ball, Dachman-Soled, and Kulkarni [2] previously constructed a NIZK for circuit satisfiability against NC^1 adversaries in the uniform random string (URS) model, where the setup only samples public coins. Their scheme achieves offline zero-knowledge, meaning that the distribution of honest URSs and proofs is computationally indistinguishable from that of the output of a simulator drawn from a

specific distribution. However, their construction is not in the fully fine-grained setting, since their prover requires more computational resources than NC^1 (even for statements represented as NC^1 circuits). This requirement is inherent in their construction, since the underlying NIZK for $\oplus\mathsf{L}/\mathsf{poly}$ in their construction requires computing the determinant of a matrix, which cannot be done in NC^1.

More recently, Wang and Pan [32] proposed a fully fine-grained NIZK protocol for circuit satisfiability in NC^1, where all algorithms (including the CRS generator, prover, verifier, and simulator) are in NC^1. Their scheme can achieve either perfect soundness or perfect zero-knowledge and can be converted into a NIZK in the URS model and a non-interactive zap. Notably, their underlying NIZK for linear languages supports the same class of languages as our QA-NIZK. However, their construction has larger proving/verification cost and proof size than ours. Especially, their proof size is dependent of the statement size, while ours is not.

Another fine-grained NIZK is recently proposed by Wang and Pan [33] in a different fine-grained setting under no assumption. Specifically, it treats adversaries in AC^0 and requires that all algorithms run in AC^0.

C Instantiations of Encodings

In this section, other than the one in Figure 3, we give several examples of predicate encodings in Figures 24, 25, and 26. By instantiating our resulting ABE in Section 5 with these encodings, we immediately achieve ABEs for inner product, non-zero inner product, and boolean span programs. All the encodings can be performed in $\mathsf{AC}^0[2]$ since they only involve multiplication of a constant number of matrices.

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

Figure 24: Definitions of the predicate and encoding of an ABE scheme for inner product (with short secret keys).

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

Figure 25: Definitions of the predicate and encoding of an ABE scheme for non-zero inner product (with short secret ciphertexts).

References

[1] Applebaum, B., Ishai, Y., Kushilevitz, E.: Cryptography in NC^0. In: 45th FOCS. pp. 166–175. IEEE Computer Society Press (Oct 2004) (Cited on page 3.)

[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) (Cited on page 2, 4, 33.)

$\mathcal{X} = \{0,1\}^n$ $\mathcal{Y} = \{0,1\}^{n \times n'}$ $\ell = n + (n' - 1)$ $\eta = n$ $\zeta = n$	$\mathsf{sE}_\lambda((\mathbf{x}, \omega), \begin{pmatrix} \mathbf{w} \\ \mathbf{u} \end{pmatrix}) = (x_1 w_1, \cdots, x_n w_n)$ $\mathsf{rE}_\lambda(\mathbf{M}, \begin{pmatrix} \mathbf{w} \\ \mathbf{u} \end{pmatrix}) = (\mathbf{M}_1 \begin{pmatrix} 0 \\ \mathbf{u} \end{pmatrix} + w_1, \cdots, \mathbf{M}_n \begin{pmatrix} 0 \\ \mathbf{u} \end{pmatrix} + w_n)$ $\mathsf{kE}_\lambda(\mathbf{y}, \alpha) = (\mathbf{M}_1 \begin{pmatrix} \alpha \\ \mathbf{0} \end{pmatrix}, \cdots, \mathbf{M}_n \begin{pmatrix} \alpha \\ \mathbf{0} \end{pmatrix})$
$\mathsf{p}_\lambda((\mathbf{x}, \omega), \mathbf{M})$: Return 1 iff $\mathbf{x}$ satisfies $\mathbf{M}$	$\mathsf{sD}_\lambda((\mathbf{x}, \omega), \mathbf{M}, \mathbf{c}) = \sum_{i=1}^n \omega_i c_i$ $\mathsf{rD}_\lambda((\mathbf{x}, \omega), \mathbf{M}, \mathbf{d}) = \sum_{i=1}^n x_i \omega_i d_i$

Figure 26: Definitions of the predicate and encoding of an ABE scheme for boolean span programs. $\mathbf{x}$ satisfies $\mathbf{M}$ w.r.t. some ω iff $\sum_{i:x_i=1} \omega_i \mathbf{M}_i = (1, 0, \cdots, 0)^\top$, where $\mathbf{M}_i$ denotes the ith row of $\mathbf{M}$. Notice that in the original definition in [10], ω is not part of the attribute and is computed from $\mathbf{M}$ and $\mathbf{x}$ in decryption. We put ω into the attribute since computing it from $\mathbf{M}$ and $\mathbf{x}$ involves Gaussian elimination which cannot be executed in NC^1. This does not affect the security of the resulting ABE since ω can be efficiently computed publicly in the original encoding scheme anyway.

[3] Barrington, D.A.M.: Bounded-width polynomial-size branching programs recognize exactly those languages in NC^1. In: 18th ACM STOC. pp. 1–5. ACM Press (May 1986) (Cited on page 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) (Cited on page 3.)

[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) (Cited on page 2, 3, 4, 5, 9, 10, 11.)

[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) (Cited on page 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) (Cited on page 2.)

[8] Brzuska, C., Couteau, G.: Towards fine-grained one-way functions from strong average-case hardness. IACR Cryptol. ePrint Arch. 2020, 1326 (2020) (Cited on page 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) (Cited on page 2.)

[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) (Cited on page 2, 3, 4, 7, 8, 19, 35.)

[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) (Cited on page 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) (Cited on page 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) (Cited on page 2, 3, 6, 31.)

[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) (Cited on page 2, 3, 4.)

[15] Egashira, S., Wang, Y., Tanaka, K.: Fine-grained cryptography revisited. J. Cryptol. 34(3), 23 (2021) (Cited on page 3, 6, 7, 28, 31.)

[16] Fuchsbauer, 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) (Cited on page 1.)

[17] Gentry, C., Silverberg, A.: Hierarchical ID-based cryptography. In: Zheng, Y. (ed.) ASIACRYPT 2002. LNCS, vol. 2501, pp. 548–566. Springer, Heidelberg (Dec 2002) (Cited on page 4.)

[18] 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 (Cited on page 2.)

[19] 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) (Cited on page 3.)

[20] 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) (Cited on page 4.)

[21] 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) (Cited on page 3.)

[22] 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) (Cited on page 2, 28.)

[23] Kiltz, E., Wee, H.: Quasi-adaptive NIZK for linear subspaces revisited. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015, Part II. LNCS, vol. 9057, pp. 101–128. Springer, Heidelberg (Apr 2015) (Cited on page 30.)

[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) (Cited on page 4.)

[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) (Cited on page 1.)

[26] Merkle, R.C.: Secure communications over insecure channels. Commun. ACM 21(4), 294–299 (1978) (Cited on page 2.)

[27] Morillo, P., Ràfols, C., Villar, J.L.: The kernel matrix Diffie-Hellman assumption. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016, Part I. LNCS, vol. 10031, pp. 729–758. Springer, Heidelberg (Dec 2016) (Cited on page 28, 29.)

[28] 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) (Cited on page 2, 5.)

[29] 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) (Cited on page 2.)

[30] 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) (Cited on page 1.)

[31] 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) (Cited on page 2, 5.)

[32] Wang, Y., Pan, J.: Non-interactive zero-knowledge proofs with fine-grained security. In: Dunkelman, O., Dziembowski, S. (eds.) EUROCRYPT 2022, Part II. LNCS, vol. 13276, pp. 305–335. Springer, Heidelberg (May / Jun 2022) (Cited on page 4, 6, 33, 34.)

[33] Wang, Y., Pan, J.: Unconditionally secure NIZK in the fine-grained setting. In: Agrawal, S., Lin, D. (eds.) ASIACRYPT 2022, Part II. LNCS, vol. 13792, pp. 437–465. Springer, Heidelberg (Dec 2022) (Cited on page 4, 33, 34.)

[34] Wang, Y., Pan, J., Chen, Y.: Fine-grained secure attribute-based encryption. In: Malkin, T., Peikert, C. (eds.) CRYPTO 2021, Part IV. LNCS, vol. 12828, pp. 179–207. Springer, Heidelberg, Virtual Event (Aug 2021) (Cited on page 1, 4.)

[35] 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) (Cited on page 4.)

[36] Wee, H.: Dual system encryption via predicate encodings. In: Lindell, Y. (ed.) TCC 2014. LNCS, vol. 8349, pp. 616–637. Springer, Heidelberg (Feb 2014) (Cited on page 2, 3, 4, 19.)