Fully Adaptive Decentralized MA-ABE: Simplified, Optimized, ASP Supported

Pratish Datta[1], Junichi Tomida[1], and Nikhil Vanjani[2]

[1]NTT Research
[2]Carnegie Mellon University

Abstract

We revisit decentralized multi-authority attribute-based encryption (MA-ABE) through the lens of fully adaptive security – the most realistic setting in which an adversary can decide on-the-fly which users and which attribute authorities to corrupt. Previous constructions either tolerated only static authority corruption or relied on highly complex "dual system with dual-subsystems" proof technique that inflated ciphertexts and keys.

Our first contribution is a streamlined security analysis showing – perhaps surprisingly – that the classic Lewko–Waters MA-ABE scheme [EUROCRYPT 2011] already achieves full adaptive security, provided its design is carefully reinterpreted and, more crucially, its security proof is re-orchestrated to conclude with an information-theoretic hybrid in place of the original target-group-based computational step. By dispensing with dual subsystems and target-group-based assumptions, we achieve a significantly simpler and tighter security proof along with a more lightweight implementation. Our construction reduces ciphertext size by 33 percent, shrinks user secret keys by 66 percent, and requires 50 percent fewer pairing operations during decryption – all while continuing to support arbitrary collusions of users and authorities. These improvements mark a notable advance over the state-of-the-art fully adaptive decentralized MA-ABE scheme of Datta et al. [EUROCRYPT 2023]. We instantiate the scheme in both composite-order bilinear groups under standard subgroup-decision assumptions and in asymmetric prime-order bilinear groups under the Matrix-Diffie–Hellman assumption. We further show how the Kowalczyk–Wee attribute-reuse technique [EUROCRYPT 2019] seamlessly lifts our construction from "one-use" boolean span programs (BSP) to "multi-use" policies computable in NC^1, resulting in a similarly optimized construction over the state-of-the-art by Chen et al. [ASIACRYPT 2023].

Going beyond the Boolean world, we present the first MA-ABE construction for arithmetic span program (ASP) access policies, capturing a richer class of Boolean, arithmetic, and combinatorial computations. This advancement also enables improved concrete efficiency by allowing attributes to be handled directly as field elements, thereby eliminating the overhead of converting arithmetic computations into Boolean representations. The construction – again presented in composite and prime orders – retains decentralization and adaptive user-key security, and highlights inherent barriers to handling corrupted authorities in the arithmetic setting.

Contents

1 Introduction **3**

2 Technical Overview **7**
- 2.1 Background on MA-ABE and Fully Adaptive Security . . . 8
- 2.2 The Lewko–Waters Construction and its Limitations in supporting Adaptive Authority Corruption 10
- 2.3 Our Approach for Supporting Adaptive Authority Corruption . . . 11
- 2.4 Our composite-order Fully Adaptive MA-ABE scheme for BSP . . . 12
- 2.5 Our Composite-order Fully Adaptive MA-ABE scheme for ASP . . . 16
- 2.6 A Compiler to support corruption of authorities related to the challenge ciphertext . . . 17

3 Notations **17**

4 MA-ABE for monotone BSP from prime-order groups **18**
- 4.1 Proof of Theorem 4.1 . . . 18

5 MA-ABE for NC^1 with Multi-Use Security **25**
- 5.1 Our Construction . . . 25

6 MA-ABE for ASP from prime-order groups **25**
- 6.1 Construction . . . 25

7 Compiler for MA-ABE for ASP: boosting security **26**

A Preliminaries: Appendix **37**
- A.1 Access Structures . . . 37
- A.2 Boolean Span Program . . . 37
- A.3 Arithmetic Span Programs . . . 37
- A.4 Secret Sharing Schemes . . . 37
- A.5 MA-ABE . . . 38
- A.6 Assumptions . . . 40

B MA-ABE for monotone BSP from prime-order groups: Appendix **42**
- B.1 Correctness of MA-ABE for monotone BSP construction . . . 42
- B.2 Missing Proofs from Section 4.1 . . . 42

C MA-ABE for NC^1 with Multi-Use Security: Appendix **49**
- C.1 Core 1-ABE . . . 49
- C.2 Proof of Theorem 5.1 . . . 50

D MA-ABE for ASP from prime-order groups: Appendix **57**
- D.1 Definition . . . 57
- D.2 Correctness of MA-ABE for ASP construction . . . 59
- D.3 Proof of Theorem 6.1 . . . 59

E Generic Compiler for MA-ABE for ASP: boosting security: Appendix **69**

F Efficiency Analysis: Detailed **71**

1 Introduction

Attribute-Based Encryption (ABE) [SW05, GPSW06] provides fine-grained control over encrypted data access by associating decryption privileges directly with user credentials (attributes). Such schemes enable authorized users possessing appropriate attributes to decrypt specific messages without revealing unintended information. Over the years, designing robust and practical ABE schemes has been a vibrant research area, leading to numerous advances that balance critical trade-offs among expressiveness, efficiency, security, and cryptographic assumptions. [BSW07, OSW07, Wat09, LOS+10, LW10, OT10, AFV11, LW11b, Wat11, LW12, OT12, Wat12, Boy13, GGH+13, GVW13, Att14, BGG+14, Wee14, CGW15, Att16, BV16, ABGW17, GKW17, CGKW18, Att19, AMY19, GWW19, KW19, Tsa19, KW20, AY20, BV20, GW20, LL20b, LL20a, GW20, AT20, Wee21a, Wee21b, Wee22, RW22, AYY22, LLL22, CW23, ARYY23, JLL23, HLL23, CW24, Wee24, DHM+24, AKY24, HLL24, WW24, Wee25, CW25]. Alongside these exciting advances, ABE has seen growing adoption in practical applications [GPSW06, ETS18, VAH23, LVV+23] and standardization efforts by NIST [PB23] and ETSI [ETS18].

Multi-Authority Attribute-Based Encryption (MA-ABE). Traditional ABE schemes mentioned above rely on a single central authority possessing a master secret key to issue attribute-based decryption keys. To address the inherent limitations of centralized trust – a critical barrier to large-scale, real-world deployment – Chase [Cha07] introduced the concept of Multi-Authority Attribute-Based Encryption (MA-ABE), which enables multiple independent authorities to concurrently manage disjoint sets of attributes. Each authority issues secret keys to users for attributes under its individual control, without requiring coordination or interaction with other authorities. Consequently, a user holding attributes from multiple authorities can decrypt ciphertexts associated with an access policy by seamlessly combining keys obtained from the relevant attribute authorities.

Fully Adaptive Security for MA-ABE. Similar to standard ABE, Multi-Authority ABE naturally demands collusion resistance against unauthorized users; however, MA-ABE introduces an additional challenge: the possibility that some attribute authorities themselves may become corrupted and collude with malicious users. Addressing this critical security aspect, prior works have formalized various corruption models, typically allowing adaptive corruption of users' secret keys while limiting authorities to static corruption—chosen by an adversary at the outset of the security game. Given the inherently decentralized and dynamic nature of MA-ABE, such assumptions seem overly restrictive and unrealistic, as they imply that adversaries must commit upfront to corrupting specific authorities even before observing any issued keys or interactions. Also, in practical scenarios, it is natural to assume authorities may join or leave dynamically over time. Acknowledging these considerations, recent research has advanced toward a more robust and realistic notion known as the fully adaptive security model, where both attribute authorities and user keys can be corrupted adaptively at any point during the security game. This strengthened security definition—which captures realistic threats and dynamic environments—is the central focus of this paper.

The State-of-the-Art in MA-ABE. Despite substantial progress in designing expressive and efficient single-authority ABE schemes with strong security guarantees, extending these desirable features to the multi-authority setting remains a challenging and delicate task. As a result, only a handful of MA-ABE constructions exist, and all suffer from limitations – whether in terms of security, efficiency, or overall complexity. After several early proposals with notable shortcomings [Cha07, LCLS08, MKE09a, CC09, MKE09b], Lewko and Waters [LW11a] introduced the

first truly decentralized MA-ABE scheme. In their construction, any party can act as an independent authority by simply publishing a public key and issuing secret keys to users for attributes under its control—without requiring any global coordination beyond an initial trusted setup. Authorities operate independently, need not be aware of each other's existence, and can join the system at any time, with no bound on the total number of authorities over the system's lifetime. Their scheme supports all access policies expressible as monotone boolean span programs (BSP) [Bei96a], offering expressive policy enforcement in a fully decentralized environment. The security of the scheme is established in the random oracle model using the powerful dual system encryption framework [Wat09, LW10, LOS+10], under subgroup-decision style assumptions in composite-order bilinear groups.

Following the foundational work of Lewko and Waters [LW11a], a number of extensions and refinements to decentralized MA-ABE was proposed. Okamoto and Takashima [OT20a] presented a construction over prime-order bilinear groups under the Decision Linear (DLIN) assumption [BBS04]. Subsequent works [RW15, AG21, Ven23, dlPVA23] improved the efficiency of pairing-based MA-ABE schemes, but at the expense of weaker security guarantees and/or reliance on less standard q-type assumptions or the Generic Group Model (GGM). Similarly, in the lattice setting, existing decentralized MA-ABE constructions [DKW21, WWW22, CLW25] continue to face limitations in expressiveness, offer weaker security models, and often rely on nonstandard knowledge-based assumptions.

Despite significant progress, all MA-ABE schemes discussed thus far fall short of supporting adaptive corruption of attribute authorities. In fact, achieving fully adaptive security in a decentralized MA-ABE setting remained an open challenge for over a decade. Strikingly, this problem represents a rare instance where standard complexity leveraging or guessing-style arguments fail—even when one tolerates sub-exponential security loss [SW05, BB11, JW16, JKK+17, KW20]. The issue is that when applied in this context, such arguments incur an exponential security loss proportional to the maximum number of authorities involved in a ciphertext. Thus, such techniques require fixing this maximum number in advance and adjusting the security parameter accordingly, severely limiting the practicality and flexibility of the resulting schemes.

Breaking this long-standing barrier, Datta et al. [DKW23] were the first to construct a fully adaptively secure decentralized MA-ABE scheme for access policies expressed as monotone BSPs. Their construction supports an arbitrary polynomial number of adaptive user key and authority corruption queries—marking a major milestone in the field. They presented two variants of their scheme: one based on composite-order bilinear groups, and a more efficient instantiation over prime-order bilinear groups [Fre10, Gui13, dlPVA22], under the Subgroup Decision and Matrix Diffie-Hellman (MDDH) assumptions, respectively. Despite this progress, their scheme inherits a key limitation from earlier MA-ABE constructions with static authority corruption under static computational assumptions —namely, the "one-use" restriction, which permits each attribute to appear at most once in any access policy. Chen et al. [CCG+23] addressed this by extending the prime-order scheme of [DKW23] to support multi-use of attributes in the context of NC1 access policies. Their enhancement integrates the celebrated multi-use technique of Kowalczyk and Wee [KW19], originally developed for centralized ABE systems, and retains security under the MDDH assumption. More recently, Garg et al. [GGL24] constructed a fully adaptively secure decentralized MA-ABE scheme for access policies representable by polynomial-depth monotone circuits, albeit with support for only bounded collusion.

Motivation. This work revisits the construction and proof techniques of Datta et al. [DKW23] with the goal of developing a fully adaptive decentralized MA-ABE scheme that supports arbitrary

collusions of users and attribute authorities, while offering improved efficiency and a significantly simpler security analysis. Although the schemes of Datta et al. [DKW23] and its follow-up by Chen et al. [CCG+23] remain the state-of-the-art in achieving fully adaptive security against arbitrary collusions under standard static assumptions, their security proofs are notably intricate. In particular, they rely on a highly sophisticated extension of the dual system encryption technique, termed the "Dual System with Dual Sub-systems". This approach constructs two parallel systems—a "main system" and a "shadow system"—and carefully injects entropy into the shadow system, which is then gradually transferred to the main system through a delicate sequence of hybrid transitions, involving both computational and information-theoretic arguments. To motivate this complexity, Datta et al. [DKW23] argued in their technical overview (Section 2.3) that earlier approaches based on the framework of Lewko and Waters [LW11a], and its many extensions, are fundamentally inadequate for handling adaptive authority corruptions. The key challenge lies in the structure of the authority master secret keys: in [LW11a], these keys are composed of exponents, which cannot be directly embedded into dual system proofs, as such proofs operate over group elements and rely on subgroup decision-style assumptions for computational indistinguishability. To overcome this limitation, Datta et al. [DKW23] introduced a pivotal modification: they replaced a portion of the authority master secret keys with a global public group element that is independent of any specific authority. This enabled the inclusion of corrupted authorities within the scope of the dual system proof framework. However, this innovation also introduced new technical hurdles, necessitating the development of their elaborate dual system with dual sub-system methodology to complete the proof.

While the techniques developed in [DKW23] were a major breakthrough—resolving a long-standing open problem—there has been surprisingly little follow-up work exploring whether simpler or more efficient approaches could achieve fully adaptive security. In particular, it remains unclear whether the use of two parallel subsystems—a central component of their proof technique—is truly necessary for supporting fully adaptive security under arbitrary collusions. Could a single-system design suffice instead? If so, this would yield not only a conceptually simplified and tighter security analysis, but also an immediate efficiency gain in both computation and communication complexity, while preserving the same strong security guarantees.

A second motivation of our work is to initiate the study of MA-ABE schemes where access policies are expressed in the arithmetic model of computation. To date, all existing MA-ABE constructions encode access policies using the Boolean model, and no scheme has been designed specifically for arithmetic representations. As a natural first step, we focus on access policies realizable by Arithmetic Span Programs (ASPs)—a powerful abstraction that captures a wide range of computations. ASPs can naturally represent various arithmetic operations such as sparse polynomial evaluation, mean, and variance, as well as combinatorial tasks like string matching, finite automata, and decision trees. Importantly, Boolean/arithmetic formulas, Boolean span programs, and Boolean/arithmetic branching programs can all be efficiently converted into ASPs with only polynomial blow-up [IW14a], making ASPs a strictly more expressive model for encoding policies. While in principle one could simulate arithmetic relations using general MA-ABE schemes for Boolean circuits [GGL24, CLW25] by translating each field operation into an equivalent Boolean sub-circuit, this approach is not scalable. While providing reasonable asymptotic efficiency in theory (e.g., via fast integer multiplication techniques [Für07]), the concrete overhead of this approach is enormous. Moreover, there are practical settings where attribute values must be treated as atomic field elements, rather than decomposed into bits, rendering such Boolean encodings inapplicable. Note that in view of similar efficiency and applicability issues with boolean computations, arithmetic variants of various important cryptographic primitives have already been considered in the last few years both within attribute-based cryptography [AIK11, PHGR13, KOS16] and beyond [DOT19, LL20a, LL20b].

Our work builds on this direction by asking: Can we design a decentralized MA-ABE scheme that natively supports access policies defined by ASPs, avoiding the inefficiencies of Boolean encodings?

Our Results

We present the following contributions in this work:

1. Fully Adaptive Decentralized MA-ABE with Simpler Proof and Improved Efficiency. We construct a fully adaptively secure decentralized MA-ABE scheme for monotone BSP access policies that resists arbitrary collusions of users and attribute authorities. Our scheme achieves significant efficiency gains over the current state-of-the-art constructions [DKW23, CCG+23], while also offering a substantially simpler security proof. In particular, our analysis avoids the intricate dual system with dual-subsystem framework introduced in [DKW23], making the security argument more streamlined and transparent. As in [DKW23], we present two instantiations of our construction: one in composite-order bilinear groups (outlined in the Technical Overview, Section 2.4) and another in asymmetric prime-order bilinear groups (detailed formally in Section 4). The former relies on the Subgroup Decision assumption, while the latter is based on the MDDH assumption. Our prime-order scheme is derived via the composite to prime order translation technique of Chen et al. [CGKW18], which builds on prime-order conversion frameworks developed in [CGW15, GDCC16]. Along the way, we further optimize the required subspace dimensions to achieve improved performance.

At a technical level, our approach fundamentally differs from [DKW23]. Rather than embedding the master secret keys of corrupted authorities into the dual system framework—as pursued in [DKW23]—we keep these keys entirely outside the dual system machinery. Our main contribution lies in carefully revisiting and restructuring the original Lewko–Waters [LW11a] security proof with only minor adjustments to their scheme architecture. Our key insight is that the primary limitation of [LW11a]—their reliance on a final *computational* transition based on an assumption analogous to Decisional Bilinear Diffie-Hellman (DBDH) [BF01], which inherently demands static knowledge of corrupted authorities—can be overcome by replacing this step with a purely *information-theoretic* transition. However, applying this idea naively fails due to excessive leakage of honest authorities' master key information in their existing hybrid argument. We resolve this by carefully restructuring their security analysis to strictly limit information leakage about honest authority master keys. This critical refinement yields a simpler and tighter fully adaptive security proof without significantly modifying the original [LW11a] construction. Please refer to Sections 2.2 to 2.4 below for details.

This yields a surprising and insightful discovery: despite their deep expertise in dual system techniques, neither the authors of [LW11a] nor [DKW23] identified that the construction in [LW11a] could, with a restructured proof, be elevated to support adaptive authority corruptions. In particular, Datta et al. explicitly argued in their technical overview (Section 2.3) that incorporating the authority master secret keys into the dual system framework was essential for achieving fully adaptive security – a barrier that motivated the development of their complex dual system with dual sub-systems methodology. Our work shows that this perceived necessity can, in fact, be circumvented entirely.

Furthermore, since our security proof does not rely on target-group-based computational assumptions, we are able to relocate all target group components from the original Lewko–Waters construction [LW11a]—both in the ciphertext and the authority public keys—into the first source group, which is significantly more compact and efficient. As we demonstrate in Table 1, our MA-ABE scheme offers substantial improvements in both size and performance over the existing fully adaptive constructions that support arbitrary collusions of users and authorities [DKW23, CCG+23]. Specifically, compared to these prior schemes, our construction achieves 33% reduction in the size of authority master secret keys and ciphertexts, and a 66% reduction in user secret key size. On

the computational side, encryption requires no pairings, and decryption involves 50% fewer pairing operations.

Since our construction closely follows [LW11a], it inherits their one-use restriction, allowing each attribute to appear at most once per access policy. To overcome this and support multi-use of attributes, we adopt the technique of Kowalczyk and Wee [KW19] – as applied by Chen et al. [CCG+23] – and integrate it with our prime-order MA-ABE scheme. This results in a fully adaptively secure decentralized MA-ABE scheme for NC^1 access policies that supports attribute reuse while achieving the same efficiency gain over [CCG+23] as our single-use scheme does over [DKW23].

2. Decentralized MA-ABE for Arithmetic Span Programs (ASPs). Our second contribution is the first decentralized MA-ABE scheme for Arithmetic Span Programs (ASPs). We present constructions in both the composite-order bilinear group setting (outlined in the Technical Overview, Section 2.5) and the prime-order setting (Section 6), with security proven under the Subgroup Decision and MDDH assumptions, respectively. The security analysis builds directly upon our proof framework developed for the BSP case. Similar to our BSP scheme, our ASP-based construction supports an unbounded number of attribute authorities, each capable of independently joining the system at any time and also tolerates adaptive corruption of arbitrarily many user secret keys. However, our security proof holds under the assumption that no authority appearing in a ciphertext is corrupted.

While the inability to ensure security for ciphertexts involving corrupted authorities may seem unsatisfying, we argue in Section 2.5 that this limitation reflects fundamental barriers inherent to the problem. At a high level, security for access structures represented by ASPs is guaranteed only when a user possesses secret keys for at most one value per attribute. However, in the decentralized MA-ABE setting, if an adversary corrupts an attribute authority, it gains the ability to issue keys for multiple values of the same attribute to the same user. This capability undermines the security of ASPs and enables unauthorized decryption—breaking the intended security guarantees. This limitation is closely related to why MA-ABE schemes cannot directly support access policies realizable by non-monotone linear secret sharing scheme (LSSS) [DKW21]. The only viable approach in that case is to first reduce the non-monotone LSSS to an equivalent monotone form and then apply a secure MA-ABE construction for monotone LSSS access policies [OT20a, DKW21].

Additionally, we provide a generic workaround that allows the authorities related to the challenge ciphertext to be corrupted, but the cost of weakening the functionality of the scheme: decryptors must have keys from all authorities featured in a ciphertext policy in order to decrypt the ciphertext. Security holds as long as either the adversary corrupts no authority appearing in the challenge ciphertext policy or for each GID queried, there exists an honest authority appearing in the challenge ciphertext policy who did not issue any secret key. Thus, the modified construction achieves the security model of Cini et al. [CLW25], but unlike [CLW25], our approach does not require a very-selective security model or bounds on number of authorities, and it supports fully adaptive queries, including corruption. The details of this modification can be found in Sections 2.6 and 7.

2 Technical Overview

In this section, we present the core technical ideas behind our work. We begin by explaining how we transform the construction and the security analysis of the classic Lewko–Waters MA-ABE scheme [LW11a] to achieve fully adaptive security—supporting both adaptive user key queries and adaptive corruption of attribute authorities. As discussed in the Introduction, this refinement leads

[1] Our MA-ABE for ASP satisfies full adaptive security with Type 1 restriction as defined in Definition D.2

Table 1: Efficiency Comparison: Comparison of fully adaptively secure decentralized MA-ABE in prime order groups. All schemes are instantiated from k-MDDH assumption in prime-order asymmetric pairing groups, with $k = 1$ (that is, Symmetric External Diffie-Hellman (SXDH) assumption). n and ℓ denote the number of rows and columns in the policy matrix $\mathbf{M}$ (and also matrix $\mathbf{N}$ in case of ASP) respectively.

(a) Communication Comparision: We omit the size of c_0 and access policy in $|\mathsf{ct}|$.

Scheme	$\lvert\mathsf{msk}_i\rvert$	$\lvert\mathsf{pk}_i\rvert$	$\lvert sk_{i,\mathsf{GID}}\rvert$	$\lvert\mathsf{ct}\rvert$	Many-use?	Policy class
DKW23 [DKW23]	$18\lvert\mathbb{Z}_q\rvert$	$6\lvert\mathbb{G}_1\rvert$	$6\lvert\mathbb{G}_2\rvert$	$12n\lvert\mathbb{G}_1\rvert$	No	monotone BSP
Section 4	$12\lvert\mathbb{Z}_q\rvert$	$6\lvert\mathbb{G}_1\rvert$	$2\lvert\mathbb{G}_2\rvert$	$8n\lvert\mathbb{G}_1\rvert$	No	monotone BSP
Section 6[1]	$24\lvert\mathbb{Z}_q\rvert$	$12\lvert\mathbb{G}_1\rvert$	$2\lvert\mathbb{G}_2\rvert$	$14n\lvert\mathbb{G}_1\rvert$	No	ASP
CCG+23 [CCG+23]	$18\lvert\mathbb{Z}_q\rvert$	$6\lvert\mathbb{G}_1\rvert$	$6\lvert\mathbb{G}_2\rvert$	$12n\lvert\mathbb{G}_1\rvert$	Yes	NC^1
Section 5	$12\lvert\mathbb{Z}_q\rvert$	$6\lvert\mathbb{G}_1\rvert$	$2\lvert\mathbb{G}_2\rvert$	$8n\lvert\mathbb{G}_1\rvert$	Yes	NC^1

(b) Computation Comparison: S denotes the set of attributes with respect to which decryption is performed. For each algorithm, we specify two values: #exponentiations, #pairings. For #exponentiations, $(\mathsf{count})_i$ denotes the count for group $\mathbb{G}_i$ where $i \in \{1, 2, T\}$.

Scheme	AuthSetup	KGen	Enc	Dec
DKW23 [DKW23]	$(18)_1, 0$	$(18)_2, 0$	$(6n\ell{+}12n{+}6\ell{-}3)_1 + (3)_T, 3$	$(\lvert S\rvert)_T, 12\lvert S\rvert$
Section 4	$(12)_1, 0$	$(7)_2, 0$	$(14n)_1 + (1)_T, 0$	$(3\lvert S\rvert)_1 + (\lvert S\rvert)_T, 6\lvert S\rvert$
Section 6	$(24)_1, 0$	$(7)_2, 0$	$(26n)_1 + (1)_T, 0$	$(9\lvert S\rvert)_1 + (\lvert S\rvert)_T, 6\lvert S\rvert$
CCG+23 [CCG+23]	$(18)_1, 0$	$(18)_2, 0$	$(18n)_1 + (6)_T, 3$	$(\lvert S\rvert)_T, 12\lvert S\rvert$
Section 5	$(12)_1, 0$	$(7)_2, 0$	$(14n)_1 + (1)_T, 0$	$(3\lvert S\rvert)_1 + (\lvert S\rvert)_T, 6\lvert S\rvert$

to a significantly efficient construction for fully adaptively secure MA-ABE schemes for monotone BSP access policies, improving upon the constructions in [DKW23, LW11a, CCG+23]. We then briefly sketch how we use the technique of Kowalczyk and Wee [KW20] to extend our scheme to support multi-use of attributes within access policies.

Finally, we outline our decentralized MA-ABE construction for ASPs and elaborate on the inherent barriers to supporting authority corruptions in this more expressive arithmetic setting.

2.1 Background on MA-ABE and Fully Adaptive Security

Our MA-ABE schemes—like all known MA-ABE constructions—assume that each user in the system is associated with a unique global identifier GID, drawn from a universe of global identifiers $\mathcal{GID} \subset \{0,1\}^*$. For simplicity of exposition, we assume in this paper (without loss of generality) that each attribute is managed by a distinct authority, allowing us to use the terms "authority" and "attribute" interchangeably. (We note that this restriction can be relaxed to support an a priori bounded number of attributes per authority [LW11a].) We denote the universe of attribute authorities by $\mathcal{AU}$.

Let us briefly recall the syntax of a decentralized MA-ABE scheme. Such a scheme consists of five core algorithms: $\mathsf{GlobalSetup}, \mathsf{AuthSetup}, \mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec}$.

- The $\mathsf{GlobalSetup}$ procedure takes as input the security parameter (in unary) and outputs global public parameters gp. All subsequent procedures rely on gp, though we may omit them in the notation when clear from context.

- The $\mathsf{AuthSetup}$ procedure can be invoked by any authority $u \in \mathcal{AU}$, generating a corresponding public-secret key pair $(\mathsf{pk}_u, \mathsf{sk}_u)$.
- Using sk_u , the authority can issue a user-specific secret key $\mathsf{sk}_{\mathsf{GID},u}$ for a user identified by global identifier $\mathsf{GID} \in \mathcal{GID}$
- At any point, using the public keys $\{\mathsf{pk}_u\}$ of selected authorities, anyone can encrypt a message msg under an access policy $(\mathbb{A}, \rho)$ to obtain a ciphertext ct.
- A user possessing a collection of secret keys $\{\mathsf{sk}_{\mathsf{GID},u}\}$ corresponding to a consistent GID can decrypt a ciphertext ct if and only if the set of attributes associated with their keys satisfies the access policy embedded in the ciphertext.

In constructions built within the random oracle model[2]—including ours —a public hash function H_1 is assumed. This function maps global identifiers GID to an appropriate domain and is specified by $\mathsf{GlobalSetup}$. In the security analysis, H_1 is modeled as a random oracle.

Fully Adaptive Security. Just like standard ABE, the security of an MA-ABE scheme demands collusion resistance, that is, no group of colluding users, none of whom is individually authorized to decrypt a ciphertext, should be able to decrypt the same when they pull their secret key components together. However, in case of MA-ABE, it is further required that collusion resistance should hold even if some of the authorities collude with the adversarial users and thereby those users can freely obtain secret keys corresponding to the attributes controlled by those corrupt authorities. Decentralized MA-ABE further allows the public and secret keys of the corrupt authorities to be generated in a malicious way and still needs collusion resistance. This is crucial since, in a decentralized MA-ABE scheme, anyone is allowed to act as an attribute authority by generating its public and secret keys locally and independently of everyone else in the system. The fully adaptive security is roughly defined by the following game:
Global Setup: The challenger generates global public parameters.
Query Phase 1: The attacker is allowed to adaptively make a polynomial number of queries of the following form: (i) *Authority Setup Query* : the challenger runs $\mathsf{AuthSetup}$ to create a public/master key pair for an authority specified by the adversary, (ii) *Secret Key Query* : the challenger runs KGen to create a secret key for a given attribute. (iii) *Authority Master Key Query* : the challenger provides the attacker the master secret key of an authority of the adversary's choice.
Challenge Phase: The adversary submits two messages $\mathsf{msg}_0, \mathsf{msg}_1$, and an access structure $\mathbb{A}$ along with a set of public keys of authorities involved in the access structure. It gets back from the challenger an encryption of one of the messages (chosen at random) with respect to the access structure. It is crucial that the adversary does not hold enough secret keys/authority master keys to decrypt a message that is encrypted with respect to the access structure.
Query Phase 2: Same as in Query Phase 1 (while making sure that the constraint from the challenge phase is not violated).
Guess: The attacker guesses which message was encrypted.

The Lewko-Waters MA-ABE schemes [LW11a] consider a much weaker definition where the adversary must commit during the Global Setup phase on the set of authorities in the system as well as on the subset of corrupted authorities. Already at that point, the private/public key pairs of all non-corrupt authorities are created by the challenger and the public keys are given to the

[2]In fact, all known MA-ABE schemes except [WWW22] are built in the random oracle model. While [WWW22] is constructed without random oracles, it relies on the evasive LWE assumption, which is a strong, knowledge-type assumption whose variants have recently come under cryptanalytic scrutiny [BÜW24, HJL25, AMYY25].

attacker. (That is, during Query Phase I and II, only queries of form: (ii) *Secret Key Query* are allowed.) The fully adaptive security definition [DKW23, CCG+23] is much more realistic given the distributed nature of MA-ABE.

2.2 The Lewko–Waters Construction and its Limitations in supporting Adaptive Authority Corruption

As with any centralized ABE scheme, a central challenge in MA-ABE is achieving collusion resistance. In standard ABE, this is typically accomplished by having a central authority—who knows the master secret key—generate user keys in such a way that the individual key components (corresponding to different attributes) are "tied together" using fresh randomness unique to each user. This user-specific randomization ensures that the components are compatible only within a given user's key, and cannot be combined across users to violate access policies. In the multi-authority setting, however, we aim to achieve two goals simultaneously: (i) *autonomous key generation*, where each authority independently issues keys, and (ii) *collusion resistance*, ensuring that unauthorized users cannot pool their keys to decrypt unauthorized ciphertexts.

This autonomy requirement fundamentally limits the applicability of traditional randomization techniques—since there is no central party coordinating the key components. In decentralized MA-ABE, each key component may originate from a distinct authority, with no shared state or even awareness of one another. To address this, all existing decentralized MA-ABE schemes (with the sole exception of [WWW22]) use the output of a public hash function applied to the user's global identifier GID as a source of deterministic "shared randomness" across authorities. This ensures that all key components issued to a given user are cryptographically tied together, while maintaining full decentralization. We now briefly describe the Lewko–Waters MA-ABE scheme for monontone BSP access policies and the core challenges it faces in supporting adaptive corruption of authorities.

The Lewko-Waters MA-ABE scheme is described as follows:

- $\mathsf{gp} = (N, \mathbb{G}, \mathbb{G}_T, e, g_1, \mathsf{H}_1 : \{0,1\}^* \to \mathbb{G})$ where $(N = p_1p_2p_3, \mathbb{G}, \mathbb{G}_T, e)$ a composite-order bilinear group with $\mathbb{G}_{p_1}, \mathbb{G}_{p_2}, \mathbb{G}_{p_3}$ being the subgroups of $\mathbb{G}$ of order p_1, p_2, p_3 respectively and $g_1 \in \mathbb{G}_{p_1}$
- $\mathsf{pk}_u = (e(g_1,g_1)^{\alpha_u}, g_1^{y_u}), \mathsf{sk}_u = (\alpha_u, y_u)$ with $\alpha_u, y_u \leftarrow \mathbb{Z}_N$ for all $u \in \mathcal{AU}$
- $\mathsf{ct} = (C_0, \{C_{1,x}, C_{2,x}, C_{3,x}\}_{x\in[\ell]})$, where $C_0 = \mathsf{msg}\cdot e(g_1,g_1)^s$, for all $x \in [\ell]$: $C_{1,x} = e(g_1,g_1)^{\lambda_x + \alpha_{\rho(x)} r_x}$, $C_{2,x} = g_1^{r_x}$, $C_{3,x} = g_1^{\omega_x + y_{\rho(x)} r_x}$ where $(\mathbf{M}, \rho)$ is the BSP access policy with $\mathbf{M} \in \mathbb{Z}_N^{\ell\times d}$, $\rho : [\ell] \to \mathcal{AU}$ is injective, $\lambda_x = \mathbf{M}_x \cdot \mathbf{v}, \omega_x = \mathbf{M}_x \cdot \mathbf{w}$, $\mathbf{v} = (s, v_2, \ldots, v_d)$, $\mathbf{w} = (0, w_2, \ldots, w_d)$, where $v_2, \ldots v_d, w_2, \ldots w_d$ are uniform random in $\mathbb{Z}_q$.
- $\mathsf{sk}_{\mathsf{GID},u} = g_1^{\alpha_u} \mathsf{H}_1(\mathsf{GID})^{y_u}$

The security proof of the Lewko–Waters construction [LW11a] relies on the dual system encryption technique originally introduced by Waters [Wat09]. In this framework, ciphertexts and keys can exist in either normal form or various semi-functional forms. Semi-functional components are not part of the actual system—they are introduced solely for the purpose of the security proof. The core idea is as follows: a normal key can correctly decrypt both normal and semi-functional ciphertexts, and similarly, a normal ciphertext can be decrypted by either normal or semi-functional keys. However, when a semi-functional key is used to decrypt a semi-functional ciphertext, decryption fails. This asymmetry is leveraged to gradually transition from the real world to a simulated one in which the adversary learns no useful information. Security is established via a sequence of indistinguishable hybrid games. The first game corresponds to the real security game, where both the ciphertext and

all user keys are in normal form. In the second game, the ciphertext is switched to semi-functional form while keeping all keys normal. Then, for an adversary making q secret key queries, the proof proceeds through q hybrids: in the kth game, the first k keys are semi-functional, and the remaining are normal. In the final game—where all keys and the challenge ciphertext are semi-functional—none of the keys are useful for decrypting the challenge ciphertext, thereby ensuring security.

The security proof of Lewko and Waters [LW11a] follows the dual system encryption framework. In the penultimate step of the proof, all user secret keys and the challenge ciphertext are in semi-functional form. The final hybrid transition switches the semi-functional challenge ciphertext from an encryption of the original message to an encryption of a random message, thereby completing the proof of security. To argue the indistinguishability of this transition, the authors rely on a target-group-based computational assumption—specifically, Assumption 4 in [LW11a], which is a variant of the DBDH assumption. More precisely, they simulate the masking secret s as abc, where $a, b, c \leftarrow \mathbb{Z}_N$ are random exponents and unknown to the simulator. To accomplish this simulation without knowing the exponents, the reduction performs two steps depending on whether the authority associated with the challenge access policy is corrupt or honest. First, the reduction disables the effect of the rows in the challenge BSP access policy matrix that correspond to corrupted authorities by selecting a vector that is orthogonal to those rows. Such a vector is guaranteed to exist, since the set of corrupted authorities must, by definition, be unauthorized. This orthogonality allows the simulator to effectively "ignore" the need for master secret keys corresponding to the corrupted authorities. For the honest authorities, the situation is even more delicate. The reduction must embed elements of the underlying hard problem instance directly into their public keys. More precisely, the reduction has to embed the term ab within the master key components α of honest authorities, so that it can simulate the ciphertext components associated with honest authorities containing the shares of s by canceling out ab in the exponent. Crucially, this embedding must be performed before any authority is created. Otherwise, if the reduction embeds the unknown exponents from the challenge problem instance into the master secret key of an authority that is later corrupted, it will be unable to provide that master secret key to the adversary – breaking the simulation. These two technical dependencies – (1) selecting an orthogonal vector for unauthorized rows and (2) embedding problem instances into the public keys of honest authorities – require the reduction to know in advance which authorities will be corrupted. As a result, the security proof can only support static corruption of authorities.

2.3 Our Approach for Supporting Adaptive Authority Corruption

We begin with the same high-level objective as Datta et al. [DKW23]: avoiding the need to simulate authority master keys based on instances of computational hardness assumptions. In their work, Datta et al. achieved this by adhering to a critical design principle throughout their security proof – ensuring that all transitions involving game conditions or information about corrupted authorities are handled via information-theoretic arguments. In particular, transitions between such adjacent hybrids are shown to be statistically close by implicitly redefining all relevant components, including the authority master secret keys, without invoking computational assumptions. Our first step is to examine why this information-theoretic paradigm could not be directly applied to the original Lewko–Waters construction [LW11a], and why the invocation of the highly sophisticated "dual system with dual subsystems" framework was deemed necessary by [DKW23]. Upon careful analysis, we observe that the Lewko–Waters proof already adheres to this theme for the most part, except in the final step, where they rely on a computational transition based on a target-group hardness assumption as explained above. A natural question that arises at this point is whether the computational transition in [LW11a] can be replaced with a purely information-theoretic step,

rather than undertaking the more elaborate task of bringing the authority master keys into the fold of dual system encryption, as pursued by [DKW23]. Notably, this alternative approach leads to a significantly simpler and tighter security analysis.

However, due to the specific structure of the Lewko–Waters scheme [LW11a] and its associated hybrid transitions, we observe that the final hybrid cannot be directly transformed into an information-theoretic transition. Roughly speaking, just prior to the final hybrid step, the challenge ciphertext and all user secret keys are in semi-functional form. In this form, both the ciphertext and secret key components contain segments in the $\mathbb{G}_{p_3}$ subgroup. Notably, the $\mathbb{G}_{p_3}$ portions of the ciphertext components $\{C_{3,x}\}$ do not encode secret shares of zero, but instead encode shares of a random, independent value. Our objective in the final information-theoretic step is to transport the entropy present in the $\{C_{3,x}\}$ components into the $\mathbb{G}_{p_3}$ segments of the $\{C_{1,x}\}$ components, thereby randomizing the shares of the masking secret s. For corrupted authorities, this can be achieved by leveraging an orthogonal vector – an information-theoretic tool also used in [DKW23]. Importantly, using this vector does not violate adaptive corruption constraints since the transition remains entirely information-theoretic. For honest authorities, however, this entropy transfer must be achieved using the p_3 components of the master secret keys, conditioned on the adversary's knowledge of user secret keys issued by those authorities. Unfortunately, in the security proof of [LW11a], the semi-functional form of the user secret keys fully reveals the p_3 components of an authority's master secret key when just two such keys are issued for distinct GIDs. To resolve this, we must redesign the $\mathbb{G}_{p_3}$ segments of user secret keys so that even a polynomial number of such keys leaks only limited information about the corresponding authority master secret keys. This, in turn, necessitates a re-orchestration of the entire sequence of hybrid transitions in [LW11a].

2.4 Our composite-order Fully Adaptive MA-ABE scheme for BSP

We start by introducing several modifications to the original construction [LW11a]. First, inspired by the approach of [DKW23], we include an element $h \leftarrow \mathbb{G}$ in the global parameters and mask the ciphertext payload msg as $\mathsf{msg} \oplus \mathsf{H}_2(e(g_1, h)^s)$, where $\mathsf{H}_2 : \mathbb{G}_T \to \{0,1\}^*$ is a universal hash function [CW77, CS02] (used as a randomness extractor [V+12]) and $s \leftarrow \mathbb{Z}_N$. In the security proof, we shift g_1 from being an element of the p_1 subgroup to an element $g_1 g_3$ in the $p_1 p_3$ subgroup. As a result, the payload becomes masked by $\mathsf{H}_2(e(g_1, h)^s \boxed{e(g_3, h)^s})$, allowing us to utilize the entropy in s mod p_3 to fully randomize the payload in the final game. Additionally, since the only step in [LW11a] that relies on a target-group-based computational assumption is the final transition – which our proof replaces with information-theoretic transition – we no longer need to embed the secret shares of s in the target group. Instead, we provide ElGamal-style encryptions of the secret shares in the source group $\mathbb{G}$, under the corresponding authority public keys. Specifically, for each row x in the BSP access structure $(\mathbf{M}, \rho)$, we include: $C_{1,x} = g_1^{\lambda_x + \alpha_{\rho(x)} r_x}, C_{2,x} = g_1^{r_x}, C_{3,x} = g_1^{\omega_x + y_{\rho(x)} r_x}$ for all rows x of the associated BSP access structure $(\mathbf{M}, \rho)$.

This transformation results in smaller ciphertexts and faster encryption, since source group elements are significantly more compact than target group elements and exponentiations are much faster there. For user secret keys, instead of generating them as $\mathsf{sk}_{\mathsf{GID},u} = g_1^{\alpha_u} \mathsf{H}_1(\mathsf{GID})^{y_u}$ — as done in the original [LW11a] construction—we define them as $\mathsf{sk}_{\mathsf{GID},u} = h^{\alpha_u} \mathsf{H}_1(\mathsf{GID})^{y_u}$. Observe that, unlike [DKW23], we do not use the element h to eliminate the α components of the authority master keys. Instead, we retain α as an integral part of the authority's secret.

Security Analysis. We now outline our security proof strategy. For simplicity of exposition in this overview, we exclude the case of malicious authorities—that is, authorities created by the adversary

itself. This simplification does not impact the validity or structure of the security argument. The formal security proofs of our constructions, presented in the main body, fully account for maliciously generated authorities and handle them accordingly.

Hyb_0: This is the real fully adaptive security game described in Section 2.1.

Hyb_1: Modify the random oracle H_1 to return random elements from $\mathbb{G}_{p_1}$. This change is indistinguishable under the subgroup decision assumption between $\mathbb{G}_{p_1}$ and $\mathbb{G}$ (Assumption 4.1 in [DKW23]). Consequently, for any uncorrupted authority u, the y_u value modulo p_2 and p_3 are *information theoretically* hidden no matter how many keys the attacker requests from the authority u.

Hyb_2: Add a $\mathbb{G}_{p_3}$ component to each part of the challenge ciphertext. That is, the components of the challenge ciphertext takes the form $C_0 = \mathsf{msg}_b \oplus \mathsf{H}_2(e(g_1,h)^s \boxed{e(g_3,h)^{s''}})$, for all $x \in [\ell]$: $C_{1,x} = g_1^{\lambda_x+\alpha_{\rho(x)}r_x} \boxed{g_3^{\lambda''_x+\alpha_{\rho(x)}r''_x}}, C_{2,x} = g_1^{r_x} \boxed{g_3^{r''_x}}, C_{3,x} = g_1^{\omega_x+y_{\rho(x)}r_x} \boxed{g_3^{\omega''_x+y_{\rho(x)}r''_x}}$, where $b \leftarrow \{0,1\}, s'', \{r''_x\}_{x\in[\ell]} \leftarrow \mathbb{Z}_N, \{\lambda''_x\}_{x\in[\ell]}, \{\omega''_x\}_{x\in[\ell]},$ are secret shares of s'' and zero with respect to the challenge BSP access policy $(\mathbf{M},\rho)$. This transition follows from the subgroup decision assumption between $\mathbb{G}_{p_1}$ and $\mathbb{G}_{p_1p_3}$ (Assumption 4.2 in [DKW23]).

Hyb_3: We change the $\mathbb{G}_{p_3}$ components of $\{C_{3,x}\}_{x\in[\ell]}$ to include shares of a random value instead of zero that is, $\{\omega''_x\}_{x\in[\ell]}$ are now shares of some $w'' \leftarrow \mathbb{Z}_N$. This is an information theoretic step relying on two facts. (1) That the attacker has no information on $y_u \pmod{p_3}$ of any uncorrupted authority u per our step in Hyb_1. The fact that $y_u \mod p_3$ is hidden (and each authority appears at most once in a ciphertext) means that $C_{3,x}$ cannot be distinguished from random in the $\mathbb{G}_{p_3}$ subgroup. Thus, the share is hidden when row x corresponds to an uncorrupted authority u. (2) That the rows of the challenge matrix $(\mathbf{M},\rho)$ associated with the corrupted authorities are unauthorized for decryption. Hence, they are insufficient for learning the value of the p_3 component of the shared secret.

Critically, this step relies on an information-theoretic argument, which eliminates the challenges associated with embedding a reduction to a computational assumption in the presence of adaptive corruptions. As highlighted earlier, this principle is a central theme of our entire reduction strategy. Throughout the proof, we separate computational and information-theoretic components. All aspects of the analysis that depend on the adversary's corruption behavior are confined to the information-theoretic segments, where adaptivity poses no difficulty.

Hyb_4: Add a $\mathbb{G}_{p_2}$ component to each part of challenge ciphertext. That is, the challenge ciphertext takes the form $C_0 = \mathsf{msg}_b \oplus \mathsf{H}_2(e(g_1,h)^s \boxed{e(g_2,h)^{s'}} e(g_3,h)^{s''})$, for all $x \in [\ell]$: $C_{1,x} = g_1^{\lambda_x+\alpha_{\rho(x)}r_x} \boxed{g_2^{\lambda'_x+\alpha_{\rho(x)}r'_x}} g_3^{\lambda''_x+\alpha_{\rho(x)}r''_x}$, $C_{2,x} = g_1^{r_x} \boxed{g_2^{r'_x}} g_3^{r''_x}, C_{3,x} = g_1^{\omega_x+y_{\rho(x)}r_x} \boxed{g_2^{\omega'_x+y_{\rho(x)}r'_x}} g_3^{\omega''_x+y_{\rho(x)}r''_x}$, where $b \leftarrow \{0,1\}, s', \{r'_x\}_{x\in[\ell]} \leftarrow \mathbb{Z}_N, \{\lambda'_x\}_{x\in[\ell]}, \{\omega'_x\}_{x\in[\ell]},$ are secret shares of s' and zero with respect to the challenge access policy $(\mathbf{M},\rho)$. This transition follows from subgroup decision assumption between $\mathbb{G}_{p_1}$ and $\mathbb{G}_{p_1p_2}$ (Assumption 4.3 in [DKW23]).

Hyb_5: Modify the random oracle H_1 to return random elements from $\mathbb{G}_{p_1p_3}$ with the restriction that the $\mathbb{G}_{p_3}$ *components of all the random oracle* H_1 *outputs being the same.* The proof that this change is indistinguishable actually goes through a sequence of sub-hybrids where we change the

oracle queries one by one. Intuitively, changing the random oracle H_1 output for a certain GID is akin to making the secret key components for GID to be semi-functional. Thus, the proof will need to leverage the fact that the key components acquired by GID do not satisfy the challenge ciphertext access structure even when combined with corrupt authorities. For each GID the proof will first establish this in the $\mathbb{G}_{p_2}$ subgroup to be "temporarily semi-functional", then use this to move it to the "permanent semi-functional" space in $\mathbb{G}_{p_3}$. While doing this latter movement, the proof will carefully ensure that the $\mathbb{G}_{p_3}$ component is the same across all the random oracle H_1 queries. Finally, undo the work in the $\mathbb{G}_{p_2}$ space to make it available for moving the next GID over.

We consider the following sequence of sub-hybrids for each query GID_j for H_1.

- First modify the random oracle output $\mathsf{H}_1(\mathsf{GID}_j)$ to be a random element in $\mathbb{G}_{p_1p_2}$ instead of $\mathbb{G}_{p_1}$. This change is indistinguishable under the subgroup decision (Assumption 4.3 in [DKW23]) between $\mathbb{G}_{p_1}$ and $\mathbb{G}_{p_1p_2}$.

- Modify the $\mathbb{G}_{p_2}$ components of $\{C_{3,x}\}_{x\in[\ell]}$ to involve shares of a random value as opposed to zero, that is, $\{\omega'_x\}_{x\in[\ell]}$ are now shares of some random $w' \leftarrow \mathbb{Z}_N$. This is an information theoretic step which uses the fact that the rows of the challenge matrix $(\mathbf{M}, \rho)$ associated with the corrupted authorities in conjunction with all those rows for which the adversary requests a secret key for GID_j are unauthorized for decryption. The adaptive corruption of the authority as well as the adaptive key requests for GID_j do not cause any problems. We emphasize that since this information theoretic argument is done over the $\mathbb{G}_{p_2}$ subgroup, it does not matter whether the adversary has information about the $\mathbb{G}_{p_3}$ from keys for other global identities. This is the benefit for modifying keys one by one in an isolated subspace.

- Next, add the same $\mathbb{G}_{p_3}$ component to $\mathsf{H}_1(\mathsf{GID}_j)$ that was added to all the prior H_1 queries. We prove the indistinguishable of this transition under a subgroup decision assumption between $\mathbb{G}_{p_1p_2}$ and $\mathbb{G}$ which is a slight variant of Assumption 4.4 of [DKW23]. More precisely, the assumption we use states that an element of the form $T_1T_2 \in \mathbb{G}_{p_1p_2}$ is indistinguishable from $T_1T_2X_3 \in \mathbb{G}$ given $((N = p_1p_2p_3, \mathbb{G}, \mathbb{G}_T, e), g_1, g_2, X_1X_3, Z_2Z_3)$, where $g_1, X_1, T_1 \leftarrow \mathbb{G}_{p_1}, g_2, Z_2, T_2 \leftarrow \mathbb{G}_{p_2}, X_3, Z_3 \leftarrow \mathbb{G}_{p_3}$. This new assumption can be proven to hold under the original Assumption 4.3 of [DKW23] in three hybrid steps[3].

- Modify the $\mathbb{G}_{p_2}$ components of $\{C_{3,x}\}_{x\in[\ell]}$ to again involve shares of zero, that is, $\{\omega'_x\}_{x\in[\ell]}$ are, once again, shares of zero. This is again an information theoretic step similar to the Hyb_4.

- Remove the $\mathbb{G}_{p_2}$ component of the random oracle output $\mathsf{H}_1(\mathsf{GID}_j)$, that is, make it a random element from $\mathbb{G}_{p_1p_3}$ with the restriction that its $\mathbb{G}_{p_3}$ component is now the same as all prior H_1 query. This transition is indistinguishable under the subgroup decision assumption between $\mathbb{G}_{p_1}$ and $\mathbb{G}_{p_1p_2}$ (Assumption 4.3 of [DKW23]).

Note that in the above sequence of sub-hybrids, the $\mathbb{G}_{p_2}$ subgroup is repeatedly used to "escort" values to the $\mathbb{G}_{p_3}$ subgroup. So far, the structure of the proof broadly mirrors that of [LW11a,DKW23],

[3]The original Assumption 4.3 of [DKW23] ensures indistinguishability between random elements of $\mathbb{G}_{p_1p_2}$ and those of $\mathbb{G}$ given the same auxiliary information as our structured variant. We can reduce our assumption to Assumption 4.3 of [DKW23] via the following hybrid steps. In the first step applying Assumption 4.3 of [DKW23], we move from a random $T_1T_2 \leftarrow \mathbb{G}_{p_1p_2}$ to a random $T_1T_2Y_3 \leftarrow \mathbb{G}$, where $Y_3 \leftarrow \mathbb{G}_{p_3}$ independent of $X_3 \in \mathbb{G}_{p_3}$. The second transition is an information-theoretic step where we visualize $T_1T_2Y_3$ as $T_1T_2(U_3X_3)$ where $U_3 \leftarrow \mathbb{G}_{p_3}$. This holds since p_3 is prime. In the third step, we move from $T_1T_2U_3X_3$ to $T_1T_2X_3$ via another application of Assumption 4.3 from [DKW23]. Here, given an instance of Assumption 4.3 of [DKW23] where the challenge element is T, we prepare our challenge element as TX_1X_3. It is easy to observe that if $T \leftarrow \mathbb{G}$, then we simulate $T_1T_2U_3X_3$ while if $T \leftarrow \mathbb{G}_{p_1p_2}$, we simulate $T_1T_2X_3$.

although there are important differences in the low-level details. In particular, unlike [LW11a], which employs a single semi-functional form of the ciphertext, our approach introduces multiple semi-functional forms to handle the more complex setting of adaptive authority corruption, in addition to adaptive secret key queries. Another critical distinction from both [LW11a] and [DKW23] lies in the treatment of $\mathsf{H}_1(\mathsf{GID}_j)$: at the end of this sub-hybrid sequence, all such outputs share the same $\mathbb{G}_{p_3}$ component. This uniformity is crucial in the information-theoretic step that follows, as it tightly bounds the amount of information revealed to the adversary about $\alpha_u \mod p_3$ and $y_u \mod p_3$ for honest authorities u via secret key queries. It is at this next step that we sharply depart from both [LW11a] and [DKW23]. The former could not support adaptive authority corruption due to reliance on a computational transition at this point, while the latter resorted to an intricate proof involving a long sequence of interleaved computational and information-theoretic steps across two parallel subsystems. By contrast, we handle this transition in a single, information-theoretic step, offering both conceptual simplicity and proof efficiency.

Hyb_6: Change the $\mathbb{G}_{p_3}$ components of $\{C_{1,x}\}_{x\in[\ell]}$ to include shares of an independent random value as opposed to the secret s'' used in the masking term of C_0. In order to show that this change is statistically indistinguishable, we argue that the game transcript after Hyb_5 is identically distributed to that in Hyb_6 once we make some implicit adjustments. First, note that the shares $\{\lambda''_x\}_{x\in[\ell]}$ and in the set $\{\omega''_x\}_{x\in[\ell]}$ for all the rows x of the challenge access matrix $\mathbf{M}$ associated with corrupt authorities are information theoretically revealed to the adversary. However, by the game restriction the subspace spanned by those rows does not include the vector $(1, 0, \ldots, 0)$. We may assume that this holds modulo p_3. This means that there must exist a vector $\mathbf{z} \in \mathbb{Z}_N^d$ such that $\mathbf{z}$ is orthogonal to all these rows of $\mathbf{M}$ but is not orthogonal to $(1, 0, \ldots, 0)$, (i.e., the first entry of $\mathbf{z}$ is nonzero). Let vector $\mathbf{v}''$ and $\mathbf{w}''$ be the vectors generating the shares $\{\lambda''_x\}_{x\in[\ell]}$ and $\{\omega''_x\}_{x\in[\ell]}$ respectively, that is, s'' be the first entry of vector $\mathbf{v}''$. Consider the vectors $\hat{\mathbf{v}}'' = \mathbf{v}'' + t\mathbf{z}$ and $\hat{\mathbf{w}}'' = \mathbf{w}'' - \frac{t}{c}\mathbf{z}$ where $t \leftarrow \mathbb{Z}_N$ and $X_3 = h_3^c$ is the $\mathbb{G}_{p_3}$ component of $\mathsf{H}_1(\mathsf{GID})$. Since the first entry of $\mathbf{z}$ is nonzero and t is uniformly and independently distributed over $\mathbb{Z}_N$, it follows that the first entry of $\hat{\mathbf{v}}''$ is uniformly and independently distributed from s''. Also, since $\mathbf{w}''$ is already uniformly distributed so becomes $\hat{\mathbf{w}}''$. Hence, it follows that if we generate the shares $\{\lambda''_x\}_{x\in[\ell]}$ and $\{\omega''_x\}_{x\in[\ell]}$ using $\hat{\mathbf{v}}''$ and $\hat{\mathbf{w}}''$ respectively, then we arrive at Hyb_6. It is sufficient to show that the entire game transcript when the shares $\{\lambda''_x\}_{x\in[\ell]}$ and $\{\omega''_x\}_{x\in[\ell]}$ are generated using $\hat{\mathbf{v}}''$ and $\hat{\mathbf{w}}''$ respectively instead is statistically close to that at the end of Hyb_5. We divide our argument into the following three cases:

1. For rows x of $\mathbf{M}$ corresponding to corrupt authorities, the shares λ''_x and ω''_x remain unaffected by this transformation, as the vector $\mathbf{z}$ is chosen specifically to be orthogonal to these rows.

2. For rows x of $\mathbf{M}$ associated with honest authorities for which no user keys have been issued, the $\mathbb{G}_{p_3}$ segments of $C_{1,x}$ and $C_{3,x}$ that are revealed to the adversary remain unchanged if we implicitly redefine the p_3 segments of the master secret keys for these authorities as $\hat{\alpha}_{\rho(x)} = \alpha_{\rho(x)} - (r''_x)^{-1}(\mathbf{M}_x \cdot t\mathbf{z})$ and $\hat{y}_{\rho(x)} = y_{\rho(x)} + (c \cdot r''_x)^{-1}(\mathbf{M}_x \cdot t\mathbf{z})$. Since no user keys are queried for these authorities, the values $\alpha_{\rho(x)} \mod p_3$ and $y_{\rho(x)} \mod p_3$ remain information-theoretically hidden from the adversary, ensuring that this implicit change remains completely unnoticed.

3. For rows x of $\mathbf{M}$ associated with honest authorities for which one or more user keys have been issued, the $\mathbb{G}_{p_3}$ segments of $C_{1,x}$ and $C_{3,x}$ that are information-theoretically revealed to the adversary remain unaltered under our implicit redefinition of the p_3 segments of the authority master keys, as previously described. However, in contrast to the previous case, the issuance of

one or more user keys (now in semi-functional form, each containing a $\mathbb{G}_{p_3}$ segment) means that the values $\alpha_{\rho(x)} \mod p_3$ and $y_{\rho(x)} \mod p_3$ are no longer information-theoretically hidden from the adversary. Nevertheless, due to our careful structuring at the end of Hyb_5 – specifically, ensuring that all H_1 queries yield identical $\mathbb{G}_{p_3}$ segments – the only information about the p_3 segments of the master secret keys of such authorities revealed to the adversary is the sum $\alpha_{\rho(x)} + c \cdot y_{\rho(x)} \mod p_3$, even when multiple keys are issued to distinct users. Crucially, our implicit redefinition satisfies $\hat{\alpha}_{\rho(x)} + c \cdot \hat{y}_{\rho(x)} \mod p_3 = \alpha_{\rho(x)} + c \cdot y_{\rho(x)} \mod p_3$. Consequently, this change remains completely indistinguishable from the adversary's perspective, even given the information available from the issued user keys.

Importantly, all the changes administered in this hybrid transition are implicit. Thus, as mentioned above, the adaptive corruption of authorities is not an issue.

Hyb_7: Replace C_0 with a random value unrelated to the message msg_b. Due to the work done so far, $s'' \mod p_3$ is information theoretically hidden and so s'' has at least $\log(p_3)$ bits of entropy. At this point, the security of the universal hash function H_2 hides the message.

Supporting Multi-use of Attributes. For enabling our scheme to support multi-use of attributes, we can use the same technique as [CCG+23] which built on the core 1-ABE framework by [KW19]. In the multi-use setting, some information theoretical steps in the above hybrid transition need to be changed into computational ones. The crux of this technique is that it allows us to guess the rows of LSSS, for which neither the corresponding authority is corrupted nor the corresponding secret key is queried. Hence, we can embed a computational problem into authority public keys without caring about adaptive corruptions and secret key queries. Specifically, some information theoretic steps in $\mathsf{Hyb}_3, \mathsf{Hyb}_5, \mathsf{Hyb}_6$ are changed to computational ones.

2.5 Our Composite-order Fully Adaptive MA-ABE scheme for ASP

Our construction of MA-ABE for BSP can be applied to arithmetic span programs. Recall that ASP is represented by two matrices and a map $(\mathbf{M}, \mathbf{N}, \rho)$ where the heights of $\mathbf{M}, \mathbf{N}$ are ℓ and $\rho : [\ell] \to \mathcal{AU}$. The ASP is satisfied by $\mathbf{z} \in \mathbb{Z}_N^S$ for $S \subset \mathcal{AU}$ if and only if $z_{\rho(x)}\mathbf{M}_x + \mathbf{N}_x$ for $x \in S$ spans $(1, 0, ..., 0)$. Informally, our construction of MA-ABE for ASP in composite-order groups is as follows:

- $\mathsf{gp} = (N, \mathbb{G}, \mathbb{G}_T, e, g_1, h, \mathsf{H}_1 : \{0,1\}^* \to \mathbb{G})$ is the same as in our scheme for Boolean LSSS.
- $\mathsf{pk}_u = g_1^{\alpha_u}, g_1^{\alpha'_u}, g_1^{y_u}, g_1^{y'_u}, \mathsf{sk}_u = (\alpha_u, \alpha'_u, y_u, y'_u)$ with $\alpha_u, y_u \leftarrow \mathbb{Z}_N$ for all $u \in \mathcal{AU}$
- $\mathsf{ct} = (C_0, \{C_{1,x}, ..., C_{5,x}\})$ where

$$C_0 = \mathsf{msg} \oplus \mathsf{H}_2(e(g_1, h)^s),\ C_{1,x} = g_1^{r_x},\ C_{2,x} = g_1^{\lambda_x + \alpha_{\rho_{(x)}} r_x},\ C_{3,x} = g_1^{\lambda'_x + \alpha'_{\rho_{(x)}} r_x}$$
$$C_{4,x} = g_1^{\omega_x + y_{\rho_{(x)}} r_x},\ C_{5,x} = g_1^{\omega'_x + y'_{\rho_{(x)}} r_x}$$

  with $\lambda_x = \mathbf{M}_x \cdot \mathbf{v},\ \lambda'_x = \mathbf{N}_x \cdot \mathbf{v},\ \omega_x = \mathbf{M}_x \cdot \mathbf{w},\ \omega'_x = \mathbf{N}_x \cdot \mathbf{w}$ for all $x \in [\ell]$, where $(\mathbf{M}, \mathbf{N}, \rho)$ is the arithmetic span program, the first element of $\mathbf{v}$ is s, the first element of $\mathbf{w}$ is 0, and ρ is injective
- $\mathsf{sk}_{\mathsf{GID},u} = h^{z_u \alpha_u + \alpha'_u} \mathsf{H}_1(\mathsf{GID})^{z_u y_u + y'_u}$ where $z_u \in \mathbb{Z}_N$ is the attribute value.

The security proof is also quite similar to our Boolean LSSS. Via a series of hybrids, ω_x and ω'_x in $\mathbb{G}_{p_3}$ are changed to the shares of the same random element, and all the secret keys are gradually changed into the semi-functional form. In the final step, we argue that $s \bmod p_3$ is information theoretically hidden to the adversary. A caveat is that we require that the authorities related to the challenge ciphertext are not corrupted in the security proof. This is because if the adversary obtains an authority secret key sk_u, it can generate a *malformed* secret key $\widetilde{\mathsf{sk}}_{\mathsf{GID},u} = h^{z_u\alpha_u}\mathsf{H}_1(\mathsf{GID})^{z_u y_u}$ for $z_u \in \mathbb{Z}_N$. The problem is that this secret key is a valid secret key that allows the adversary to incorporate the row $z_{\rho(x)}\mathbf{M}_x$ into the ASP computation in decryption, while only the rows of the form $z_{\rho(x)}\mathbf{M}_x + \mathbf{N}_x$ are supposed to be valid. This may allow the adversary to decrypt the challenge ciphertext without violating admissibility.

Preventing this kind of attack seems quite challenging. Intuitively, the authority should have the power to generate two secret keys for the same GID with different values, i.e., $h^{z_u\alpha_u+\alpha'_u}\mathsf{H}_1(\mathsf{GID})^{z_u y_u+y'_u}$ and $h^{z'_u\alpha_u+\alpha'_u}\mathsf{H}_1(\mathsf{GID})^{z'_u y_u+y'_u}$ for any $z_u, z'_u \in \mathbb{Z}_N$ s.t. $z_u \neq z'_u$. However, one divided by the other results in the malformed key. Hence, we need to remove this homomorphic structure from secret keys to avoid this attack, retaining the information theoretic security property for the security proof. It is unclear how we can overcome this challenge.

2.6 A Compiler to support corruption of authorities related to the challenge ciphertext

While we do not have a solution for the above problem, we present a generic workaround that allows the authorities related to the challenge ciphertext to be corrupted, but the cost of weakening the functionality of the scheme: decryptors must possess keys from all authorities featured in a ciphertext policy in order to decrypt the ciphertext. Security holds as long as either the adversary corrupts no authority appearing in the challenge ciphertext policy or for each GID queried, there exists an honest authority appearing in the challenge ciphertext policy who did not issue any secret key. Thus, the modified construction achieves the security model of Cini et al. [CLW25], but unlike [CLW25], our approach does not require a very-selective security model or bounds on number of authorities, and it supports fully adaptive queries, including corruption.

Informally, the modified construction is obtained via a compiler consisting of two layers: an inner layer of MA-ABE for ASP with full adaptive security subject to the above caveat, and an outer layer of MA-ABE for conjunctions with full adaptive security (implied by MA-ABE for monotone BSP). In the new scheme, to encrypt a message under an ASP policy, we encrypt the message with inner scheme, then encrypt the result under policy "conjunction of all authorities in set U" using outer scheme, where U is set of authorities appearing in the ASP policy. Decryption uses all keys from set U to first decrypt outer ciphertext, recovering the inner one, which is then decrypted using the ASP scheme. The security proof proceeds by guessing at the outset whether the adversary's challenge will avoid corrupted authorities. If yes, we reduce the security to that of inner ASP scheme; if no, to the conjunction scheme. For full details, we refer to Section 7.

3 Notations

Throughout, we use λ to denote the security parameter. Boldface letters such as $\mathbf{x}$ denote column vectors, and normal-font letters such as x to denote scalars. Let $\mathbf{A}$ denote a matrix over $\mathbb{Z}_q$. We use $\mathsf{span}(\mathbf{A})$ to denote the column span of $\mathbf{A}$, and we use $\mathsf{span}^m(\mathbf{A})$ to denote matrices with width m where each column lies in $\mathsf{span}(\mathbf{A})$; this means $\mathbf{M} \xleftarrow{\$} \mathsf{span}^m(\mathbf{A})$ is a random matrix m where each column is chosen uniformly from $\mathsf{span}(\mathbf{A})$. We use $\mathsf{basis}(\mathbf{A})$ to denote a basis of $\mathsf{span}(\mathbf{A})$.

Notation for pairing groups. Let PGGen be a probabilistic polynomial time algorithm that takes input a security parameter 1^λ and outputs a pairing group description $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, q, g_1, g_2, g_T)$ where $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ are groups of prime order q and $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ defines a pairing operation. Further, g_1, g_2 are random generator in $\mathbb{G}_1, \mathbb{G}_2$ respectively and $g_T := e(g_1, g_2)$. As a shorthand, the notation $[\![x]\!]_i$ means g_i^x for $i \in \{1, 2, T\}$. For a vector $\mathbf{x} = (x_1, \ldots, x_n)^T$, for $i \in \{1, 2, T\}$, $[\![\mathbf{x}]\!]_i$ denotes a vector of n group elements $(g_i^{x_1}, \ldots, g_i^{x_n})^T$. Similarly, for a matrix $\mathbf{A} = (\mathbf{a}_1, \ldots, \mathbf{a}_\ell)$ consisting of ℓ vectors, $[\![\mathbf{A}]\!]_i$ denotes a matrix of group elements of dimensions $n \times \ell$: $([\![\mathbf{a}_1]\!]_i, \ldots, [\![\mathbf{a}_\ell]\!]_i)$. For two matrices $\mathbf{A}$ and $\mathbf{B}$ of appropriate dimensions, define $e([\![\mathbf{A}]\!]_1, [\![\mathbf{B}]\!]_2) := [\![\mathbf{AB}]\!]_T$. Further, we denote group operations with $\cdot$, i.e., $[\![\mathbf{A}]\!]_i \cdot [\![\mathbf{B}]\!]_i = [\![\mathbf{A} + \mathbf{B}]\!]_i$. Also, we define $\mathbf{B} \odot [\![\mathbf{A}]\!]_i = [\![\mathbf{BA}]\!]_i$ and $[\![\mathbf{A}]\!]_i \odot \mathbf{B} = [\![\mathbf{AB}]\!]_i$.

For distributions D_0 and D_1, we write $D_0 \approx_c D_1$ if they are computationally indistinguishable and $D_0 \approx_s D_1$ if they are statistically indistinguishable.

4 MA-ABE for monotone BSP from prime-order groups

Let $\mathcal{AU}$ denote the authority universe and let each authority control one attribute. Let $\mathcal{GID}$ denote the universe of global identifiers of the users. We construct MA-ABE for access policies specified by a monotone boolean span program (BSP) denoted by a pair $(\mathbf{M}, \rho)$ of a matrix $\mathbf{M} \in \mathbb{Z}_q^{n \times \ell}$ and a labeling function $\rho : [n] \to U$, where $U \subseteq \mathcal{AU}$ denotes a subset of authorities. Our construction is as in Figure 1.

Correctness. We provide the formal proof of correctness in Appendix B.1. Here, we provide an informal sketch. Let's look at the three components of the partial decryptions d_x for all $\rho(x) \in S$ where S is the set of attributes possessed by the decrypter. $e(c_{2,x}, [\![\mathbf{k}]\!]_2)$ transforms an encryption of λ_x under the public key $[\![\mathbf{AV}_{\rho(x)}]\!]_1$ to an encryption of $\lambda_x \mathbf{k}$ under the key $[\![\mathbf{AV}_{\rho(x)}\mathbf{k}]\!]_T$. Similarly, $e(c_{3,x}, \mathsf{H}_1(\mathsf{GID}))$ transforms an encryption of ω_x under the public key $[\![\mathbf{AU}_{\rho(x)}]\!]_1$ to an encryption of $\omega_x \mathbf{h}_{\mathsf{GID}}$ under the key $[\![\mathbf{AU}_{\rho(x)}\mathbf{h}_{\mathsf{GID}}]\!]_T$, where $[\![\mathbf{h}_{\mathsf{GID}}]\!]_2 := \mathsf{H}_1(\mathsf{GID})$. The product of these two terms is thus an encryption of $\lambda_x \mathbf{k} + \omega_x \mathbf{h}_{\mathsf{GID}}$ under the key $[\![\mathbf{A}(\mathbf{V}_{\rho(x)}\mathbf{k} + \mathbf{U}_{\rho(x)}\mathbf{h}_{\mathsf{GID}})]\!]_T$. Finally, $e(c_{1,x}, \mathsf{sk}_{\rho(x),\mathsf{GID}})$ transforms secret key $\mathsf{sk}_{\rho(x),\mathsf{GID}} = [\![\mathbf{V}_{\rho(x)}\mathbf{k} + \mathbf{U}_{\rho(x)}\mathbf{h}_{\mathsf{GID}}]\!]_2$ to a secret key $[\![\mathbf{A}(\mathbf{V}_{\rho(x)}\mathbf{k} + \mathbf{U}_{\rho(x)}\mathbf{h}_{\mathsf{GID}})]\!]_T$. Thus, d_x is essentially a partial decryption resulting in $[\![\lambda_x \mathbf{k} + \omega_x \mathbf{h}_{\mathsf{GID}}]\!]_T$. Since λ_x are secret shares of $\mathbf{t}^T$ and ω_x are secret shares of $\mathbf{0}^T$, thus if the policy is satisfied by S, then the final value d is $[\![\mathbf{t}^T\mathbf{k} + \mathbf{0}^T\mathbf{h}_{\mathsf{GID}}]\!]_T = [\![\mathbf{t}^T\mathbf{k}]\!]_T$. Thus, the decryption outputs $\mathsf{msg} = c_0/d$.

Theorem 4.1. *The MA-ABE construction for monotone BSP in Figure 1 is fully adaptively secure (Definition A.6). if all of the following hold true.*

- *game condition holds and ρ is injective.*
- *k-MDDH assumption holds in groups $\mathbb{G}_1$ and $\mathbb{G}_2$ (Assumption A.8).*
- $\mathsf{SD}^{\mathbb{G}_2}_{\mathbf{B}_1 \to \mathbf{B}_1, \mathbf{B}_2}$ *assumption holds (Assumption A.9).*
- $\mathsf{SD}^{\mathbb{G}_2}_{\mathbf{B}_2 \to \mathbf{B}_2, \mathbf{B}_3}$ *assumption holds (Assumption A.9).*

We prove Theorem 4.1 in Section 4.1.

4.1 Proof of Theorem 4.1

We prove full adaptive security of the MA-ABE scheme for monotone BSP presented in Figure 1 subject to the one-use restriction, that is, ρ is injective. We prove via a sequence of hybrid games.

$\mathsf{GlobalSetup}(1^\lambda)$:

1 : $\mathcal{PG} := (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, q, g_1, g_2, g_T)$ $\leftarrow \mathsf{PGGen}(1^\lambda)$

2 : $\mathbf{A}_1 \xleftarrow{\$} \mathbb{Z}_q^{k\times(k+1)}$, $\mathbf{k} \xleftarrow{\$} \mathbb{Z}_q^{2k+1}$

3 : Sample hash function: $\mathsf{H}_1 : \mathcal{GID} \to \mathbb{G}_2^{2k+1}$

4 : **ret** $\mathsf{gp} := (\mathcal{PG}, [\![\mathbf{A}_1]\!]_1, \mathbf{k}, \mathsf{H}_1)$

$\mathsf{AuthSetup}(\mathsf{gp}, i)$:

1 : $\mathbf{V}_i, \mathbf{U}_i \xleftarrow{\$} \mathbb{Z}_q^{(k+1)\times(2k+1)}$

2 : $\mathsf{pk}_{i,0} := [\![\mathbf{A}_1\mathbf{V}_i]\!]_1, \mathsf{pk}_{i,1} := [\![\mathbf{A}_1\mathbf{U}_i]\!]_1$

3 : **ret** $\mathsf{msk}_i := (\mathbf{V}_i, \mathbf{U}_i)$, $\mathsf{pk}_i := (\mathsf{pk}_{i,0}, \mathsf{pk}_{i,1})$

$\mathsf{Enc}(\mathsf{gp}, \mathsf{msg} \in \mathbb{G}_T, (\mathbf{M}, \rho), \{\mathsf{pk}_{\rho(i)}\}_{i\in[n]})$:

1 : $\mathbf{t} \xleftarrow{\$} \mathbb{Z}_q^{2k+1}$, $\mathbf{T}_{\mathsf{bot}} \xleftarrow{\$} \mathbb{Z}_q^{(\ell-1)\times(2k+1)}$, $\mathbf{T} := \begin{pmatrix} \mathbf{t}^T \\ \mathbf{T}_{\mathsf{bot}} \end{pmatrix}$

2 : $\mathbf{W}_{\mathsf{bot}} \xleftarrow{\$} \mathbb{Z}_q^{(\ell-1)\times(2k+1)}$, $\mathbf{W} := \begin{pmatrix} \mathbf{0}^T \\ \mathbf{W}_{\mathsf{bot}} \end{pmatrix}$

3 : $\forall x \in [n]$: let $\lambda_x := \mathbf{M}_x\mathbf{T} \in \mathbb{Z}_q^{1\times(2k+1)}$, $\omega_x := \mathbf{M}_x\mathbf{W} \in \mathbb{Z}_q^{1\times(2k+1)}$

4 : $c_0 := \mathsf{msg} \cdot [\![\mathbf{t}^T\mathbf{k}]\!]_T \in \mathbb{G}_T$

5 : $\forall x \in [n]$: $\mathbf{s}_x \xleftarrow{\$} \mathbb{Z}_q^k$, $c_{1,x} := [\![\mathbf{s}_x^T\mathbf{A}_1]\!]_1 \in G_1^{1\times(k+1)}$

6 : $\forall x \in [n]$: $c_{2,x} := [\![\lambda_x]\!]_1 \cdot (\mathbf{s}_x^T \odot \mathsf{pk}_{\rho(x),0}) \in \mathbb{G}_1^{1\times(2k+1)}$

7 : $\forall x \in [n]$: $c_{3,x} := [\![\omega_x]\!]_1 \cdot (\mathbf{s}_x^T \odot \mathsf{pk}_{\rho(x),1}) \in \mathbb{G}_1^{1\times(2k+1)}$

8 : **ret** $\mathsf{ct} := (c_0, \{c_{1,x}, c_{2,x}, c_{3,x}\}_{x\in[n]})$

$\mathsf{KGen}(\mathsf{gp}, \mathsf{msk}_i, \mathsf{GID})$:

1 : Compute $[\![\mathbf{h}_{\mathsf{GID}}]\!]_2 := \mathsf{H}_1(\mathsf{GID})$

2 : **ret** $\mathsf{sk}_{i,\mathsf{GID}} := [\![\mathbf{V}_i\mathbf{k} + \mathbf{U}_i\mathbf{h}_{\mathsf{GID}}]\!]_2$

$\mathsf{Dec}(\mathsf{gp}, (\mathbf{M}, \rho), \mathsf{ct}, \mathsf{GID}, \{\mathsf{sk}_{\rho(x),\mathsf{GID}}\}_{\rho(x)\in S})$:

1 : If $(1, 0, \ldots, 0) \notin \mathit{RowSpan}(\mathbf{M}_x)$: **ret** $\perp$

2 : Let $\{w_x\}_{\rho(x)\in S}$ be constants s.t. $\sum_{\rho(x)\in S} w_x\mathbf{M}_x = (1, 0, \ldots, 0)$

3 : $\forall \rho(x) \in S$: $d_x := e(c_{2,x}, [\![\mathbf{k}]\!]_2) \cdot \frac{e(c_{3,x}, \mathsf{H}_1(\mathsf{GID}))}{e(c_{1,x}, \mathsf{sk}_{\rho(x),\mathsf{GID}})}$

4 : Compute $d := \prod_{\rho(x)\in S} d_x^{w_x}$

5 : **ret** c_0/d

Figure 1: Construction: MA-ABE for monotone BSP from prime-order groups

Suppose the adversary makes Q queries to the random oracle H_1. The hybrid games are as follows: $\mathsf{Hyb}_{\mathsf{Real}}, \mathsf{Hyb}'_{\mathsf{Real}}, \mathsf{Hyb}_1, \mathsf{Hyb}_2, \{\mathsf{Hyb}_{3,j,1}, \mathsf{Hyb}'_{3,j,1}, \mathsf{Hyb}_{3,j,2}, \mathsf{Hyb}'_{3,j,2}, \mathsf{Hyb}_{3,j,3}\}_{j\in[Q]}, \mathsf{Hyb}_4, \mathsf{Hyb}_5$.

Hybrid $\mathsf{Hyb}_{\mathsf{Real}}$. This is the real-world game $\mathsf{MA\text{-}ABE}_{\mathcal{A}}^{\mathsf{fully\text{-}adaptive}}$.

Hybrid $\mathsf{Hyb}'_{\mathsf{Real}}$. This is same as $\mathsf{Hyb}_{\mathsf{Real}}$ except that the challenger computes $\boxed{(\mathsf{gp}, \mathsf{st}) \leftarrow \mathsf{GlobalSetup}^*(1^\lambda)}$ and provides gp to the adversary. Here, $\mathsf{GlobalSetup}^*$ runs the same computation as $\mathsf{GlobalSetup}$ to compute gp and additionally also computes the following:

$$\mathbf{A}_2 \xleftarrow{\$} \mathbb{Z}_q^{1\times(k+1)}, \mathbf{B}_1, \mathbf{B}_2 \xleftarrow{\$} \mathbb{Z}_q^{(2k+1)\times k}, \mathbf{B}_3 \xleftarrow{\$} \mathbb{Z}_q^{(2k+1)\times 1}$$
$$(\mathbf{A}_1^*, \mathbf{A}_2^*) = \begin{pmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \end{pmatrix}^{-1}, (\mathbf{B}_1^*, \mathbf{B}_2^*, \mathbf{B}_3^*) = \left((\mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3)^{-1}\right)^T.$$

Then, $\mathsf{st} = (\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_1^*, \mathbf{A}_2^*, \mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3, \mathbf{B}_1^*, \mathbf{B}_2^*, \mathbf{B}_3^*)$. Observe that $\forall i, j \in \{1, 2\}$: $\mathbf{A}_i\mathbf{A}_j^* = \mathbf{I}$ if $i = j$, and $\mathbf{0}$ if $i \neq j$, and $\forall i, j \in \{1, 2, 3\}$: $\mathbf{B}_i^T\mathbf{B}_j^* = \mathbf{I}$ if $i = j$, and $\mathbf{0}$ if $i \neq j$.

Hybrid Hyb_0. This is same as $\mathsf{Hyb}_{\mathsf{Real}}$ except that the hash function H_1 is programmed to output all hash values in $\mathsf{span}(\mathbf{B}_1)$ as follows: on input GID, sample $\mathbf{h}_{\mathsf{GID}} \xleftarrow{\$} \boxed{\mathbb{Z}_q^k}$ and output $\mathsf{H}_1(\mathsf{GID}) = [\![\boxed{\mathbf{B}_1\mathbf{h}_{\mathsf{GID}}}]\!]_2$.

Hybrid Hyb_1. This is same as Hyb_0 except that the ciphertext is changed to semi-functional form. Specifically, let $U_{\mathcal{A}}$ denote the rows of $\mathbf{M}$ that correspond to the public keys provided by the adversary $\mathcal{A}$, that is $U_{\mathcal{A}} = \{i \in [n] : \rho(i) \in U_{\mathcal{A}}\}$. Let $\overline{U_{\mathcal{A}}} = [n] \setminus U_{\mathcal{A}}$. Then, we can write the normal ciphertexts as follows: $\mathsf{ct} := (c_0, \{c_{1,x}, c_{2,x}, c_{3,x}\}_{x\in[n]})$, where $c_0 := \mathsf{msg}_b \cdot [\![\mathbf{t}^T\mathbf{k}]\!]_T$, $\forall x \in [n]$: $c_{1,x} := [\![\mathbf{c}_x^T]\!]_1 := [\![\mathbf{s}_x^T\mathbf{A}_1]\!]_1$, $c_{2,x} := [\![\lambda_x]\!]_1 \cdot (\mathbf{s}_x^T \odot \mathsf{pk}_{\rho(x),0})$ if $x \in U_{\mathcal{A}}$, else $c_{2,x} := [\![\lambda_x]\!]_1 \cdot (\boxed{\mathbf{c}_x^T \odot [\![\mathbf{V}_{\rho(x)}]\!]_1})$, $c_{3,x} := [\![\omega_x]\!]_1 \cdot (\mathbf{s}_x^T \odot \mathsf{pk}_{\rho(x),1})$ if $x \in U_{\mathcal{A}}$, else $c_{3,x} := [\![\omega_x]\!]_1 \cdot (\boxed{\mathbf{c}_x^T \odot [\![\mathbf{U}_{\rho(x)}]\!]_1})$. Given this notation, the ciphertext in Hyb_1 is same as the normal ciphertext except that for all $x \in \overline{U_{\mathcal{A}}}$: $\boxed{c_{1,x} := [\![\mathbf{c}_x^T]\!]_1}$, where $\boxed{\mathbf{c}_x \xleftarrow{\$} \mathbb{Z}_q^{k+1}}$. We call this semi-functional ciphertext. Observe that in total this means that semi-functional ciphertext changes $c_{1,x}, c_{2,x}, c_{3,x}$ for $x \in \overline{U_{\mathcal{A}}}$.

Hybrid Hyb_2. This is same as Hyb_1 except that the ciphertext structure is changed as follows: the first row of matrix $\mathbf{W}$ is changed from $\mathbf{0}^T$ to $\gamma\mathbf{B}_3^{*T}$, where $\gamma \xleftarrow{\$} \mathbb{Z}_q$, that is, $\mathbf{W} = \begin{pmatrix} \boxed{\gamma\mathbf{B}_3^{*T}} \\ \hline \mathbf{W}_{\mathsf{bot}} \end{pmatrix}$.

Hybrid $\mathsf{Hyb}_{3,j-1}$ for $j \in [q+1]$. This hybrid is same as Hyb_2 except that for the i^{th} global identifier GID_i for $i \leq j-1$, the challenger programs the output $\mathsf{H}_1(\mathsf{GID}_i)$ of the random oracle H_1 as $\mathsf{H}_1(\mathsf{GID}_i) = [\![\boxed{\mathbf{B}_1\mathbf{h}_{\mathsf{GID}_i} + \mathbf{B}_3}]\!]_2$, where $\mathbf{h}_{\mathsf{GID}_i} \xleftarrow{\$} \mathbb{Z}_q^k$, while for $i > j-1$, it programs the output $\mathsf{H}_1(\mathsf{GID}_i)$ of the random oracle H_1 as $\mathsf{H}_1(\mathsf{GID}_i) = [\![\mathbf{B}_1\mathbf{h}_{\mathsf{GID}_i}]\!]_2$ as earlier.

Observe that $\mathsf{Hyb}_{3,0}$ is same as Hyb_2. We introduce a sequnce of intermediate hybrids $\mathsf{Hyb}_{3,j,1}$, $\mathsf{Hyb}'_{3,j,1}, \mathsf{Hyb}_{3,j,2}, \mathsf{Hyb}'_{3,j,2}, \mathsf{Hyb}_{3,j,3}$ between $\mathsf{Hyb}_{3,j-1}$ and $\mathsf{Hyb}_{3,j}$ for all $j \in [q]$ as defined below.

Hybrid $\mathsf{Hyb}_{3,j,1}$ for $j \in [Q]$. This hybrid is same as $\mathsf{Hyb}_{3,j-1}$ except that for the j^{th} global identifier GID_j, the challenger programs the output $\mathsf{H}_1(\mathsf{GID}_j)$ of the random oracle H_1 as $\mathsf{H}_1(\mathsf{GID}_j) = [\![\boxed{\mathbf{B}_1\mathbf{h}_{\mathsf{GID}_j} + \mathbf{B}_2\mathbf{h}'_{\mathsf{GID}_j}}]\!]_2$, where $\mathbf{h}_{\mathsf{GID}_j}, \mathbf{h}'_{\mathsf{GID}_j} \xleftarrow{\$} \mathbb{Z}_q^k$.

Hybrid $\mathsf{Hyb}'_{3,j,1}$. This is same as $\mathsf{Hyb}_{3,j,1}$ except that the ciphertext structure is changed as follows: the first row of matrix $\mathbf{W}$ is changed from $\gamma\mathbf{B}_3^{*T}$ to $(\mathbf{B}_2^*\boldsymbol{\delta} + \gamma\mathbf{B}_3^*)^T$, where $\gamma \xleftarrow{\$} \mathbb{Z}_q$ and $\boldsymbol{\delta} \xleftarrow{\$} \mathbb{Z}_q^k$ that is, $\mathbf{W} = \begin{pmatrix} \boxed{(\mathbf{B}_2^*\boldsymbol{\delta} + \gamma\mathbf{B}_3^*)^T} \\ \hline \mathbf{W}_{\mathsf{bot}} \end{pmatrix}$.

Hybrid $\mathsf{Hyb}_{3,j,2}$. This is same as $\mathsf{Hyb}'_{3,j,1}$ except that for the j^{th} global identifier GID_j, the challenger programs the output $\mathsf{H}_1(\mathsf{GID}_j)$ of the random oracle H_1 as $\mathsf{H}_1(\mathsf{GID}_j) = [\![\boxed{\mathbf{B}_1\mathbf{h}_{\mathsf{GID}_j} + \mathbf{B}_2\mathbf{h}'_{\mathsf{GID}_j} + \mathbf{B}_3}]\!]_2$, where $\mathbf{h}_{\mathsf{GID}_j}, \mathbf{h}'_{\mathsf{GID}_j} \xleftarrow{\$} \mathbb{Z}_q^k$.

Hybrid $\mathsf{Hyb}'_{3,j,2}$. This is same as $\mathsf{Hyb}_{3,j,2}$ except that the ciphertext structure is changed as follows: the first row of matrix $\mathbf{W}$ is changed from $(\mathbf{B}_2^*\boldsymbol{\delta} + \gamma\mathbf{B}_3^*)^T$ to $\gamma\mathbf{B}_3^{*T}$, where and $\gamma \xleftarrow{\$} \mathbb{Z}_q$, that is, $\mathbf{W} = \begin{pmatrix} \boxed{\gamma\mathbf{B}_3^{*T}} \\ \hline \mathbf{W}_{\mathsf{bot}} \end{pmatrix}$.

Hybrid Hyb_4. This is same as hybrid $\mathsf{Hyb}_{3,Q}$ except that the ciphertext structure is changed as follows: the first row of matrix $\mathbf{T}$ is changed from $\mathbf{t}^T$ to $(\mathbf{t} + \tau\mathbf{B}_3^*)^T$, where $\tau \xleftarrow{\$} \mathbb{Z}_q$, that is $\mathbf{T} = \begin{pmatrix} \boxed{(\mathbf{t} + \tau\mathbf{B}_3^*)^T} \\ \hline \mathbf{T}_{\mathsf{bot}} \end{pmatrix}$. Crucially, we note that c_0 remains unchanged, that is, the masking term is still $[\![\mathbf{t}^T\mathbf{k}]\!]_T$.

Hybrid Hyb_5. This is same as Hyb_4 except that the ciphertext is changed to an encryption of a random value, that is, c_0 is changed from $c_0 := \mathsf{msg}_b \cdot [\![\mathbf{t}^T\mathbf{k}]\!]_T$ to $c_0 := \zeta \cdot [\![\mathbf{t}^T\mathbf{k}]\!]_T$, where $\zeta \xleftarrow{\$} \mathbb{G}_T$.

Claim 4.2. *Hybrids $\mathsf{Hyb}_{\mathsf{Real}}$ and $\mathsf{Hyb}'_{\mathsf{Real}}$ are identically distributed.*

Claim 4.3. *If $\mathsf{MDDH}^{\mathbb{G}_2}_{k,2k+1}$ holds, then $\mathsf{Hyb}'_{\mathsf{Real}} \approx_c \mathsf{Hyb}_0$.*

Claim 4.4. *If $\mathsf{MDDH}^{\mathbb{G}_1}_{k,k+1}$ holds, then $\mathsf{Hyb}_0 \approx_c \mathsf{Hyb}_1$.*

Claim 4.5. *If game condition holds and ρ is injective, then, $\mathsf{Hyb}_1 \approx_s \mathsf{Hyb}_2$.*

Claim 4.6. *If $\mathsf{SD}^{\mathbb{G}_2}_{\mathbf{B}_1\to\mathbf{B}_1,\mathbf{B}_2}$ holds, then $\mathsf{Hyb}_{3,j-1} \approx_c \mathsf{Hyb}_{3,j,1}$ for all $j \in [Q]$.*

Claim 4.7. *If game condition holds and ρ is injective, then, $\mathsf{Hyb}_{3,j,1} \approx_s \mathsf{Hyb}'_{3,j,1}$.*

Claim 4.8. *If $\widetilde{\mathsf{SD}}^{\mathbb{G}_2}_{\mathbf{B}_2\to\mathbf{B}_2,\mathbf{B}_3}$ holds , then $\mathsf{Hyb}'_{3,j,1} \approx_c \mathsf{Hyb}_{3,j,2}$ for all $j \in [Q]$.*

Here, we note that the $\widetilde{\mathsf{SD}}^{\mathbb{G}_2}_{\mathbf{B}_2\to\mathbf{B}_2,\mathbf{B}_3}$ assumption is a variant of the Subgroup Decision assumption. We formally state this assumption below and show that it reduces to the original Subgroup Decision assumption in Claim 4.16.

Claim 4.9. *If game condition holds and ρ is injective, then, $\mathsf{Hyb}_{3,j,2} \approx_s \mathsf{Hyb}'_{3,j,2}$.*

Claim 4.10. *If $\mathsf{SD}^{\mathbb{G}_2}_{\mathbf{B}_1\to\mathbf{B}_1,\mathbf{B}_2}$ holds , then $\mathsf{Hyb}'_{3,j,2} \approx_c \mathsf{Hyb}_{3,j}$ for all $j \in [Q]$.*

Claim 4.11. *If game condition holds and ρ is injective, then, $\mathsf{Hyb}_{3,Q} \approx_s \mathsf{Hyb}_4$.*

Claim 4.12. $\mathsf{Hyb}_4 \approx_s \mathsf{Hyb}_5$.

Claim 4.13. *In Hyb_5, adversary $\mathcal{A}$'s winning advantage is 0.*

Thus, Claims 4.2 to 4.13 and hybrid argument imply that Theorem 4.1 holds.

In the rest of this section, we present the $\widetilde{\mathsf{SD}}^{\mathbb{G}_2}_{\mathbf{B}_2\to\mathbf{B}_2,\mathbf{B}_3}$ assumption and the proofs of Claims 4.11 and 4.12 since they are the most non-trivial ones when compared to previous work. Proofs of other claims are deferred to Appendix B.2.

We start by presenting the $\widetilde{\mathsf{SD}}^{\mathbb{G}_2}_{\mathbf{B}_2\to\mathbf{B}_2,\mathbf{B}_3}$ assumption. It is defined with respect to following set of matrices: Fix parameters ℓ_1, ℓ_2, ℓ_3. Pick random

$$\mathbf{B}_1 \xleftarrow{\$} \mathbb{Z}_q^{\ell\times\ell_1}, \mathbf{B}_2 \xleftarrow{\$} \mathbb{Z}_q^{\ell\times\ell_2}, \mathbf{B}_3 \xleftarrow{\$} \mathbb{Z}_q^{\ell\times\ell_3},$$

where $\ell := \ell_1 + \ell_2 + \ell_3$. Let $(\mathbf{B}_1^*, \mathbf{B}_2^*, \mathbf{B}_3^*)^T = (\mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3)^{-1}$ so that $\mathbf{B}_i^T\mathbf{B}_i^* = \mathbf{I}$ (known as *non-degeneracy*) and $\mathbf{B}_i^T\mathbf{B}_j^* = \mathbf{0}$ if $i \neq j$ (known as *orthogonality*).

Assumption 4.14 (Subgroup Decision Assumption Variant $\widetilde{\mathsf{SD}}^{\mathbb{G}_2}_{\mathbf{B}_2\to\mathbf{B}_2,\mathbf{B}_3}$). *Fix parameter $\ell_3 = 1$, that is $\mathbf{B}_3 \in \mathbb{Z}_q^{\ell}$, where $\ell = \ell_1 + \ell_2 + 1$. The $\widetilde{\mathsf{SD}}^{\mathbb{G}_2}_{\mathbf{B}_2\to\mathbf{B}_2,\mathbf{B}_3}$ assumptions states that for any p.p.t. adversary $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\cdot)$ such that for any security parameter $\lambda \in \mathbb{N}$,*

$$\mathsf{Adv}^{\mathsf{SD}^{\mathbb{G}_2}_{\mathbf{B}_i\to\mathbf{B}_i,\mathbf{B}_j}}_{\mathcal{A}}(\lambda) := |\Pr[\mathcal{A}(\mathcal{D}, [\![\mathbf{t}_0]\!]_2) = 1] - \Pr[\mathcal{A}(\mathcal{D}, [\![\mathbf{t}_1]\!]_2) = 1]| \leq \mathsf{negl}(\lambda)$$

where

$$\mathcal{PG} := (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, q, g_1, g_2, g_T) \leftarrow \mathsf{PGGen}(1^\lambda),$$
$$\mathcal{D} = (\mathcal{PG}, [\![\mathbf{B}_1]\!]_2, [\![\mathbf{B}_2]\!]_2, [\![\mathbf{B}_3]\!]_2, \mathsf{basis}(\mathbf{B}_1^*), \mathsf{basis}(\mathbf{B}_2^*), \mathsf{basis}(\mathbf{B}_2^*, \mathbf{B}_3^*)),$$
$$\mathbf{t}_0 := \mathbf{B}_2\mathbf{h}, \mathbf{t}_1 := \mathbf{B}_2\mathbf{h} + \mathbf{B}_3, \mathbf{h} \xleftarrow{\$} \mathbb{Z}_q^{\ell_2}.$$

Remark 4.15. *Unlike the standard Subgroup Decision assumption where $\mathbf{t}_1 := \mathbf{B}_2\mathbf{h} + r\mathbf{B}_3$ where $r \xleftarrow{\$} \mathbb{Z}_q$, here the coefficient of $\mathbf{B}_3$ is fixed to be 1. Note that this assumption is crucially used for changing the hash function output to be of the form $\mathbf{B}_1\mathbf{h} + \mathbf{B}_2\mathbf{h}' + \mathbf{B}_3$. Eventually, the final semi-functional form of the secret key will have hash output of the form $\mathbf{B}_1\mathbf{h} + \mathbf{B}_3$ and this fixed (non-random) $\mathbf{B}_3$ term is crucial for proving Claim 4.11 which is the main technical component of this work.*

Claim 4.16. *If $\mathsf{SD}^{\mathbb{G}_2}_{\mathbf{B}_2\to\mathbf{B}_2,\mathbf{B}_3}$ (Assumption A.9) holds, then $\widetilde{\mathsf{SD}}^{\mathbb{G}_2}_{\mathbf{B}_2\to\mathbf{B}_2,\mathbf{B}_3}$ (Assumption 4.14) holds.*

Proof. We will instantiate the $\mathsf{SD}^{\mathbb{G}_2}_{\mathbf{B}_2\to\mathbf{B}_2,\mathbf{B}_3}$ assumption with $\ell_3 = 1$. Then in this assumption, $\mathbf{t}_0 \leftarrow \mathsf{span}(\mathbf{B}_2)$ can be equivalently written as $\mathbf{t}_0 = \mathbf{B}_2\mathbf{h}$ for some $\mathbf{h} \xleftarrow{\$} \mathbb{Z}_q^{\ell_2}$. Also, $\mathbf{t}_1 \leftarrow \mathsf{span}(\mathbf{B}_2, \mathbf{B}_3)$ can be equivalently written as $\mathbf{t}_1 = \mathbf{B}_2\mathbf{h} + r\mathbf{B}_3$ for some $\mathbf{h} \xleftarrow{\$} \mathbb{Z}_q^{\ell_2}, r \xleftarrow{\$} \mathbb{Z}_q$. To prove the claim, we will consider a sequence of four hybrid distributions $\mathsf{Hyb}_0, \mathsf{Hyb}_1, \mathsf{Hyb}_2, \mathsf{Hyb}_4$ containing $(\mathcal{D}, [\![\mathbf{t}]\!]_2)$, where $\mathbf{t}$ is equal to the following across the hybrids: $\mathbf{B}_2\mathbf{h}$ in Hyb_0, $\mathbf{B}_2\mathbf{h} + r\mathbf{B}_3$ in Hyb_1, $\mathbf{B}_2\mathbf{h} + (r+1)\mathbf{B}_3$ in Hyb_2, and $\mathbf{B}_2\mathbf{h} + \mathbf{B}_3$ in Hyb_3. From $\mathsf{SD}^{\mathbb{G}_2}_{\mathbf{B}_2\to\mathbf{B}_2,\mathbf{B}_3}$ assumption, we know that Hyb_0 and Hyb_1 are computationally indistinguishable. Next, observe that r and $r+1$ are identically distributed for a uniform random r. Then, by post-processing lemma it follows that Hyb_1 and Hyb_2 are identically distributed.

Lastly, we argue that Hyb_2 and Hyb_3 are computationally indistinguishable assuming the $\mathsf{SD}^{\mathbb{G}_2}_{\mathbf{B}_2\to\mathbf{B}_2,\mathbf{B}_3}$ assumption. For this, we can create a simple reduction that obtains $(\mathcal{D}, [\![\mathbf{t}]\!]_2)$ from the $\mathsf{SD}^{\mathbb{G}_2}_{\mathbf{B}_2\to\mathbf{B}_2,\mathbf{B}_3}$ assumption challenger and sends $(\mathcal{D}, [\![\mathbf{t}]\!]_2 \cdot [\![\mathbf{B}_3]\!]_2)$ to the adversary. If $\mathbf{t} = \mathbf{B}_2\mathbf{h}$, then the adversary sees Hyb_3 distribution and if $\mathbf{t} = \mathbf{B}_2\mathbf{h} + r\mathbf{B}_3$, then the adversary sees Hyb_2 distribution. □

Proof of Claim 4.11. Observe that the only difference between $\mathsf{Hyb}_{3,Q}$ and Hyb_4 is in ciphertext components $c_{2,x}$ for all $x \in [n]$. Observe that $c_{2,x}$ contains $[\![\lambda_x]\!]_1$, where λ_x is a secret share of $\mathbf{t}^T \in \mathbb{Z}_q^{1\times(2k+1)}$ in $\mathsf{Hyb}_{3,Q}$, but it is a secret share of $(\mathbf{t} + \tau\mathbf{B}_3^*)^T$ in Hyb_4. Therefore, to prove that the hybrids are statistically indistinguishable, we will argue that $\tau\mathbf{B}_3^{*T}$ is information theoretically hidden to the adversary $\mathcal{A}$ in Hyb_4.

Suppose the challenge access policy $(\mathbf{M}, \rho)$ is defined over a set of authorities $U \subseteq \mathcal{AU}$, that is, $\rho : [n] \to U$. Recall from Appendix A.5 that the game condition requires that $U_{\mathcal{A}} \cap U_{\mathcal{B}} = \emptyset$ and for each $\mathsf{GID} \in \mathcal{GID}$, it is required that $S \cup S_{\mathsf{GID}} \notin (\mathbf{M}, \rho)$.

To show that $\tau\mathbf{B}_3^{*T}$ is information theoretically hidden from the adversary $\mathcal{A}$ in Hyb_4, we only need to rely on $S \notin (\mathbf{M}, \rho)$ which is implied by the second game condition. Here, $S \notin (\mathbf{M}, \rho)$ is a shorthand for $(1, 0, \ldots, 0) \notin \mathsf{rowSpan}(\{\mathbf{M}_x\}_{\rho(x)\in S})$.

Note that the vectors $\mathbf{M}_x\mathbf{T}$ for all rows x of the challenge access matrix $\mathbf{M}$ labeled by corrupt authorities (that is, $\rho(x) \in S$) are information theoretically revealed to $\mathcal{A}$. However, by the game condition the subspace spanned by those rows does not include the vector $(1, 0, \ldots, 0)$. This means that there must exist some vector $\mathbf{u}^T \in \mathbb{Z}_q^{1\times\ell}$ such that $\mathbf{u}^T$ is orthogonal to all these rows of $\mathbf{M}$ (that is, $\mathbf{M}_x\mathbf{u} = 0$) but is not orthogonal to $(1, 0, \ldots, 0)$, that is, the first entry of $\mathbf{u}^T$ must be non-zero. We consider a basis $\mathbb{U}$ of $\mathbb{Z}_q^\ell$ involving the vector $\mathbf{u}$ and write $\mathbf{T} = \begin{pmatrix} (\mathbf{t} + \tau\mathbf{B}_3^*)^T \\ \mathbf{T}_{\mathsf{bot}} \end{pmatrix} = \widetilde{\mathbf{U}} + \mathbf{u}\mathbf{b}^T$ for some $\mathbf{b} \in \mathbb{Z}_q^{2k+1}$ and some $\widetilde{\mathbf{U}} \in \mathbb{Z}_q^{\ell\times(2k+1)}$ such that each column of $\widetilde{\mathbf{U}}$ lies in the column span of $\mathbb{U} \setminus \mathbf{u}$. Hence, $\widetilde{\mathbf{U}}$ reveals no information about $\mathbf{b}$. Now since the first entry of $\mathbf{u}$ is non-zero, it follows that the first row of $\mathbf{T}$, that is, $(\mathbf{t} + \tau\mathbf{B}_3^*)^T$, depends on $\mathbf{b}$. But $\mathbf{M}_x\mathbf{T}$ for all the corrupted rows of $\mathbf{M}$ contains no information about $\mathbf{b}$ since $\mathbf{u}$ is orthogonal to all these rows. Thus, it follows that these rows do not leak information of $(\mathbf{t} + \tau\mathbf{B}_3^*)^T$.

Therefore, the only possible way for $\mathcal{A}$ to get information about $\tau\mathbf{B}_3^{*T}$ is through the ciphertext components $c_{2,x}$ corresponding to the uncorrupted rows of $\mathbf{M}$. However, for each such row x, $\mathcal{A}$ can only recover $\mathbf{c}_x^T$, $\mathbf{M}_x\mathbf{T} + \mathbf{c}_x^T\mathbf{V}_{\rho(x)}$ information theoretically. Without loss of generality, we can compute $\mathbf{V}_{\rho(x)} := \mathbf{V}_{\rho(x),1}\mathbf{B}_1^{*T} + \mathbf{V}_{\rho(x),2}\mathbf{B}_2^{*T} + \mathbf{V}_{\rho(x),3}\mathbf{B}_3^{*T}$, where $\mathbf{V}_{\rho(x),1}, \mathbf{V}_{\rho(x),2} \xleftarrow{\$} \mathbb{Z}_q^{(k+1)\times k}$, and $\mathbf{V}_{\rho(x),3} \xleftarrow{\$} \mathbb{Z}_q^{(k+1)\times 1}$. Let the first entry of $\mathbf{M}_x$ be m_x, that is, $\mathbf{M}_x = (m_x, \ldots)$. Then, observe that we can write

$$
\begin{aligned}
&\mathbf{M}_x \begin{pmatrix} (\mathbf{t} + \tau\mathbf{B}_3^*)^T \\ \mathbf{T}_{\mathsf{bot}} \end{pmatrix} + \mathbf{c}_x^T\mathbf{V}_{\rho(x)} \\
&= \mathbf{M}_x \begin{pmatrix} \mathbf{t}^T \\ \mathbf{T}_{\mathsf{bot}} \end{pmatrix} + m_x\tau\mathbf{B}_3^{*T} + \mathbf{c}_x^T\mathbf{V}_{\rho(x)} \\
&= \mathbf{M}_x \begin{pmatrix} \mathbf{t}^T \\ \mathbf{T}_{\mathsf{bot}} \end{pmatrix} + (m_x\tau + \mathbf{c}_x^T\mathbf{V}_{\rho(x),3})\mathbf{B}_3^{*T} + \mathbf{c}_x^T\mathbf{V}_{\rho(x),1}\mathbf{B}_1^{*T} + \mathbf{c}_x^T\mathbf{V}_{\rho(x),2}\mathbf{B}_2^{*T} \\
&= \mathbf{M}_x \begin{pmatrix} \mathbf{t}^T \\ \mathbf{T}_{\mathsf{bot}} \end{pmatrix} + (\mathbf{c}_x^T\mathbf{V}'_{\rho(x),3})\mathbf{B}_3^{*T} + \mathbf{c}_x^T\mathbf{V}_{\rho(x),1}\mathbf{B}_1^{*T} + \mathbf{c}_x^T\mathbf{V}_{\rho(x),2}\mathbf{B}_2^{*T}
\end{aligned}
$$

where we can write $\mathbf{V}'_{\rho(x),3} = \mathbf{V}_{\rho(x),3} + \Delta$ such that $m_x\tau = \mathbf{c}_x^T\Delta$. Therefore, to complete the proof, it suffices to argue that $\mathbf{V}_{\rho(x),3}$ and $\mathbf{V}'_{\rho(x),3}$ are identically distributed. We show this next.

Observe that since ρ is injective, hence it follows that $\mathbf{V}_{\rho(x)}$ is a fresh random matrix and the only other place it appears is in secret keys $\mathsf{sk}_{\rho(x),\mathsf{GID}}$. Specifically, $\mathsf{sk}_{\rho(x),\mathsf{GID}}$ information theoretically reveals $\mathbf{V}_{\rho(x)}\mathbf{k} + \mathbf{U}_{\rho(x)}\mathbf{B}_1\mathbf{h}_{\mathsf{GID}} + \mathbf{U}_{\rho(x)}\mathbf{B}_3$. Since $\mathbf{k}$ is uniform random, we can equivalently write it as $\mathbf{k} := \mathbf{B}_1\mathbf{k}_1 + \mathbf{B}_2\mathbf{k}_2 + k_3\mathbf{B}_3$ for uniform random $\mathbf{k}_1 \xleftarrow{\$} \mathbb{Z}_q^k$, $\mathbf{k}_2 \xleftarrow{\$} \mathbb{Z}_q^k$, $k_3 \xleftarrow{\$} \mathbb{Z}_q$. Further, we can write $\mathbf{U}_{\rho(x)} := \mathbf{U}_{\rho(x),1}\mathbf{B}_1^{*T} + \mathbf{U}_{\rho(x),2}\mathbf{B}_2^{*T} + \mathbf{U}_{\rho(x),3}\mathbf{B}_3^{*T}$, where $\mathbf{U}_{\rho(x),1}, \mathbf{U}_{\rho(x),2} \xleftarrow{\$} \mathbb{Z}_q^{(k+1)\times k}$, and $\mathbf{U}_{\rho(x),3} \xleftarrow{\$} \mathbb{Z}_q^{(k+1)\times 1}$. Then observe that we can write

$$
\begin{aligned}
&\mathbf{V}_{\rho(x)}\mathbf{k} + \mathbf{U}_{\rho(x)}\mathbf{B}_1\mathbf{h}_{\mathsf{GID}} + \mathbf{U}_{\rho(x)}\mathbf{B}_3 \\
&\quad= \mathbf{V}_{\rho(x),1}\mathbf{k}_1 + \mathbf{V}_{\rho(x),2}\mathbf{k}_2 + k_3\mathbf{V}_{\rho(x),3} + \mathbf{U}_{\rho(x),1}\mathbf{h}_{\mathsf{GID}} + \mathbf{U}_{\rho(x),3} \\
&\quad= \mathbf{V}_{\rho(x),1}\mathbf{k}_1 + \mathbf{V}_{\rho(x),2}\mathbf{k}_2 + k_3\boxed{\mathbf{V}'_{\rho(x),3}} + \mathbf{U}_{\rho(x),1}\mathbf{h}_{\mathsf{GID}} + \boxed{\mathbf{U}'_{\rho(x),3}}
\end{aligned}
$$

where we can write $\mathbf{V}'_{\rho(x),3} = \mathbf{V}_{\rho(x)} + \Delta$ and $\mathbf{U}'_{\rho(x)} = \mathbf{U}_{\rho(x)} - k_3\Delta$, where Δ is as defined above, that is, choose Δ such that $m_x\tau = \mathbf{c}_x^T\Delta$. Therefore, $\mathbf{V}_{\rho(x),3}$ and $\mathbf{V}'_{\rho(x),3}$ are identically distributed as long as $\mathbf{U}_{\rho(x),3}$ and $\mathbf{U}'_{\rho(x),3}$ are identically distributed. We show this next. Observe that since ρ

is injective, hence it follows that $\mathbf{U}_{\rho(x)}$ is a fresh random matrix and other than $\mathsf{sk}_{\rho(x),\mathsf{GID}}$, the only other place it appears is in ciphertext components $c_{3,x}$. For ciphertext components $c_{3,x}$ corresponding to the uncorrupted rows of $\mathbf{M}$, $\mathcal{A}$ can recover $\mathbf{M}_x\mathbf{W} + \mathbf{c}_x^T\mathbf{U}_{\rho(x)}$ information theoretically. Observe that we can write

$$\begin{aligned}
&\mathbf{M}_x \begin{pmatrix} \gamma\mathbf{B}_3^{*T} \\ \mathbf{W}_{\mathsf{bot}} \end{pmatrix} + \mathbf{c}_x^T\mathbf{U}_{\rho(x)} \\
&= \mathbf{M}_x \begin{pmatrix} \gamma\mathbf{B}_3^{*T} \\ \mathbf{W}_{\mathsf{bot}} \end{pmatrix} + \mathbf{c}_x^T\mathbf{U}_{\rho(x),3}\mathbf{B}_3^{*T} + \mathbf{c}_x^T\mathbf{U}_{\rho(x),2}\mathbf{B}_2^{*T} + \mathbf{c}_x^T\mathbf{U}_{\rho(x),1}\mathbf{B}_1^{*T} \\
&= \mathbf{M}_x \begin{pmatrix} \gamma\mathbf{B}_3^{*T} \\ \mathbf{W}_{\mathsf{bot}} \end{pmatrix} + \mathbf{c}_x^T(k_3\Delta)\mathbf{B}_3^{*T} + \mathbf{c}_x^T\mathbf{U}'_{\rho(x),3}\mathbf{B}_3^{*T} + \dots \\
&= \mathbf{M}_x \begin{pmatrix} \gamma\mathbf{B}_3^{*T} \\ \mathbf{W}_{\mathsf{bot}} \end{pmatrix} + m_x k_3\tau\mathbf{B}_3^{*T} + \mathbf{c}_x^T\mathbf{U}'_{\rho(x),3}\mathbf{B}_3^{*T} + \dots \\
&= \mathbf{M}_x \begin{pmatrix} (\gamma + k_3\tau)\mathbf{B}_3^{*T} \\ \mathbf{W}_{\mathsf{bot}} \end{pmatrix} + \mathbf{c}_x^T\mathbf{U}'_{\rho(x)} \\
&= \mathbf{M}_x \begin{pmatrix} \gamma'\mathbf{B}_3^{*T} \\ \mathbf{W}_{\mathsf{bot}} \end{pmatrix} + \mathbf{c}_x^T\mathbf{U}'_{\rho(x)}
\end{aligned}$$

where we can write $\gamma' = \gamma + k_3\tau$. Therefore to complete this part of the proof, it suffices to argue that γ and γ' are identically distributed. This holds true because γ, k_3, τ are uniform random, that is, $\gamma, k_3, \tau \xleftarrow{\$} \mathbb{Z}_q$.

To complete the proof, we argue that substituting $(\mathbf{U}_{\rho(x),3}, \mathbf{V}_{\rho(x),3}, \gamma)$ with $(\mathbf{U}'_{\rho(x),3}, \mathbf{V}'_{\rho(x),3}, \gamma')$ (as described above) for all rows x of matrix $\mathbf{M}$ for which the challenger sampled the authority keys (that is, uncorrupted rows plus the rows for which the adversary queried the master secret key) allows us to move from Hyb_4 to $\mathsf{Hyb}_{3,Q}$. We have already argued that this substitution does not change the distribution of the secret keys and ciphertext obtained by the adversary $\mathcal{A}$ for the uncorrupted rows of $\mathbf{M}$. For the case of rows x of $\mathbf{M}$ for which the adversary queried the master secret key, the adversary additionally learns $\mathbf{U}'_{\rho(x),3}, \mathbf{V}'_{\rho(x),3}$ and $\mathbf{M}_x \begin{pmatrix} \gamma'\mathbf{B}_3^{*T} \\ \mathbf{W}_{\mathsf{bot}} \end{pmatrix}$ in $\mathsf{Hyb}_{3,Q}$ and $\mathbf{U}_{\rho(x),3}, \mathbf{V}_{\rho(x),3}$ and $\mathbf{M}_x \begin{pmatrix} \gamma\mathbf{B}_3^{*T} \\ \mathbf{W}_{\mathsf{bot}} \end{pmatrix}$ in Hyb_4 and we argue that this does not help the adversary $\mathcal{A}$ to distinguish between $\mathsf{Hyb}_{3,Q}$ and Hyb_4. This is because $\mathbf{U}_{\rho(x),3}$ and $\mathbf{U}'_{\rho(x),3}$ are identically distributed, $\mathbf{V}_{\rho(x),3}$ and $\mathbf{V}'_{\rho(x),3}$ are identically distributed, and $\mathbf{M}_x \begin{pmatrix} \gamma\mathbf{B}_3^{*T} \\ \mathbf{W}_{\mathsf{bot}} \end{pmatrix} = \mathbf{M}_x \begin{pmatrix} \gamma'\mathbf{B}_3^{*T} \\ \mathbf{W}_{\mathsf{bot}} \end{pmatrix} = \mathbf{M}_x \begin{pmatrix} \mathbf{0}^T \\ \mathbf{W}_{\mathsf{bot}} \end{pmatrix}$ due to the game condition.

Therefore, it follows that $\mathsf{Hyb}_{3,Q}$ and Hyb_4 are statistically indistinguishable. This completes the proof of Claim 4.11. □

Proof of Claim 4.12. Observe that in Hyb_4 and Hyb_5, $\mathbf{t}$ is sampled randomly as $\mathbf{t} \xleftarrow{\$} \mathbb{Z}_q^{2k+1}$. Alternately, we can also sample it as $\mathbf{t} := \mathbf{B}_1^*\mathbf{t}_1 + \mathbf{B}_2^*\mathbf{t}_2 + t_3\mathbf{B}_3^*$, where $\mathbf{t}_1 \xleftarrow{\$} \mathbb{Z}_q^k, \mathbf{t}_2 \xleftarrow{\$} \mathbb{Z}_q^k, t_3 \xleftarrow{\$} \mathbb{Z}_q$. Note that the distribution of $\mathbf{t}$ is identical in both of the above ways. So, we will use the latter form to show that Hyb_4 and Hyb_5 are statistically indistinguishable. Observe that in both Hyb_4 and Hyb_5 the vector $\mathbf{t}$ shows up in ciphertext components c_0 and $c_{2,x}$. In $c_{2,x}$, it shows up in matrix $\mathbf{T}$ used for computing λ_x. And recall that $\mathbf{T} = \begin{pmatrix} (\mathbf{t} + \tau\mathbf{B}_3^*)^T \\ \mathbf{T}_{\mathsf{bot}} \end{pmatrix}$. Hence, the first row of $\mathbf{T}$ can be written as $(\mathbf{B}_1^*\mathbf{t}_1 + \mathbf{B}_2^*\mathbf{t}_2 + (t_3 + \tau)\mathbf{B}_3^*)^T$. Since $\tau \xleftarrow{\$} \mathbb{Z}_q$, it follows that $c_{2,x}$ reveals no information about $t_3 \in \mathbb{Z}_q$.

So, we will use the entropy of t_3 to argue that c_0 can be switch from encoding msg_b in Hyb_4 to encoding a random ζ in Hyb_5.

Without loss of generality, suppose $\mathbf{k}$ is of the form $\mathbf{k} = \mathbf{B}_1\mathbf{k}_1 + \mathbf{B}_2\mathbf{k}_2 + k_3\mathbf{B}_3$ for some $\mathbf{k}_1, \mathbf{k}_2, k_3$. Then observe that $[\![\mathbf{t}^T\mathbf{k}]\!]_T = [\![\mathbf{t}_1^T\mathbf{k}_1]\!]_T \cdot [\![\mathbf{t}_2^T\mathbf{k}_2]\!]_T \cdot [\![t_3k_3]\!]_T$. It follows then that $[\![t_3k_3]\!]_T$ is uniform random in $\mathbb{G}_T$ as long as $k_3 \neq 0$. Consequently, $\mathsf{msg}_b \cdot [\![t_3k_3]\!]_T$ and $\zeta \cdot [\![t_3k_3]\!]_T$ are identically distributed. Thus, c_0 in Hyb_4 and Hyb_5 are identically distributed.

Therefore, it follows that Hyb_4 and Hyb_5 are statistically indistinguishable, where the statistical security loss comes from the scenario when $k_3 = 0$. This completes the proof of Claim 4.12. □

5 MA-ABE for NC^1 with Multi-Use Security

Our construction of MA-ABE in Section 4 only supprts single-use of attributes since its security proof relies on ρ being injective. In this section, we present our construction of decentralized MA-ABE for NC^1 with multi-use security. To prove security, we rely on the Core 1-ABE construction of [KW19]. Due to space constrains, we defer the security proof and the Core 1-ABE needed for it to Appendix C.

5.1 Our Construction

The construction is same as the one in Section 4 except that the dimensions of some of the matrices are changed as follows:

- $\mathsf{GlobalSetup}(1^\lambda)$: $\mathbf{A}_1 \xleftarrow{\$} \mathbb{Z}_q^{k \times 2k}$.
- $\mathsf{AuthSetup}(\mathsf{gp}, i)$: $\mathbf{V}_i, \mathbf{U}_i \xleftarrow{\$} \mathbb{Z}_q^{2k \times (2k+1)}$. While this increases the size of msk_i, we note that the size of pk_i remains unchanged.
- $\mathsf{KGen}(\mathsf{gp}, \mathsf{msk}_i, \mathsf{GID})$: Change in size of msk_i affects the size of $sk_{i,\mathsf{GID}}$ as follows: $sk_{i,\mathsf{GID}} := [\![\mathbf{V}_i\mathbf{k} + \mathbf{U}_i\mathbf{h}_{\mathsf{GID}}]\!]_2 \in \mathbb{G}_2^{2k}$.
- $\mathsf{Enc}(\mathsf{gp}, \mathsf{msg}, (\mathbf{M}, \rho), \{\mathsf{pk}_{\rho(i)}\}_{i\in[n]})$: Change in size of $\mathbf{A}_1$ affects the size of $c_{1,x}$ as follows: $c_{1,x} := [\![\mathbf{s}_x^T\mathbf{A}_1]\!]_1 \in \mathbb{G}_1^{1\times 2k}$.

Theorem 5.1 (Informal)**.** *The MA-ABE construction in Figure 1 amended with dimension changes specified in Section 5.1 supports* NC^1 *circuits and is fully adaptively secure (Definition A.6).*

6 MA-ABE for ASP from prime-order groups

In this section, we present a MA-ABE construction for arithmetic span programs (ASPs). This construction generalizes the MA-ABE construction for monotone BSPs presented in Figure 1.

6.1 Construction

Let $\mathcal{AU}$ denote the authority universe and let each authority control an attribute that can be assigned values in $\mathbb{Z}_q$. Let $\mathcal{GID}$ denote the universe of global identifiers of the users. We construct MA-ABE for access policies specified by an arithmetic span program (ASP) denoted by a tuple $(\mathbf{M}, \mathbf{N}, \rho)$ of matrices $\mathbf{M}, \mathbf{N} \in \mathbb{Z}_q^{n\times\ell}$ and a labeling function $\rho : [n] \to U$, where $U \subseteq \mathcal{AU}$ denotes a subset of attributes. Our construction is as in Figure 2. We prove Theorem 6.1 in Appendix D.3.

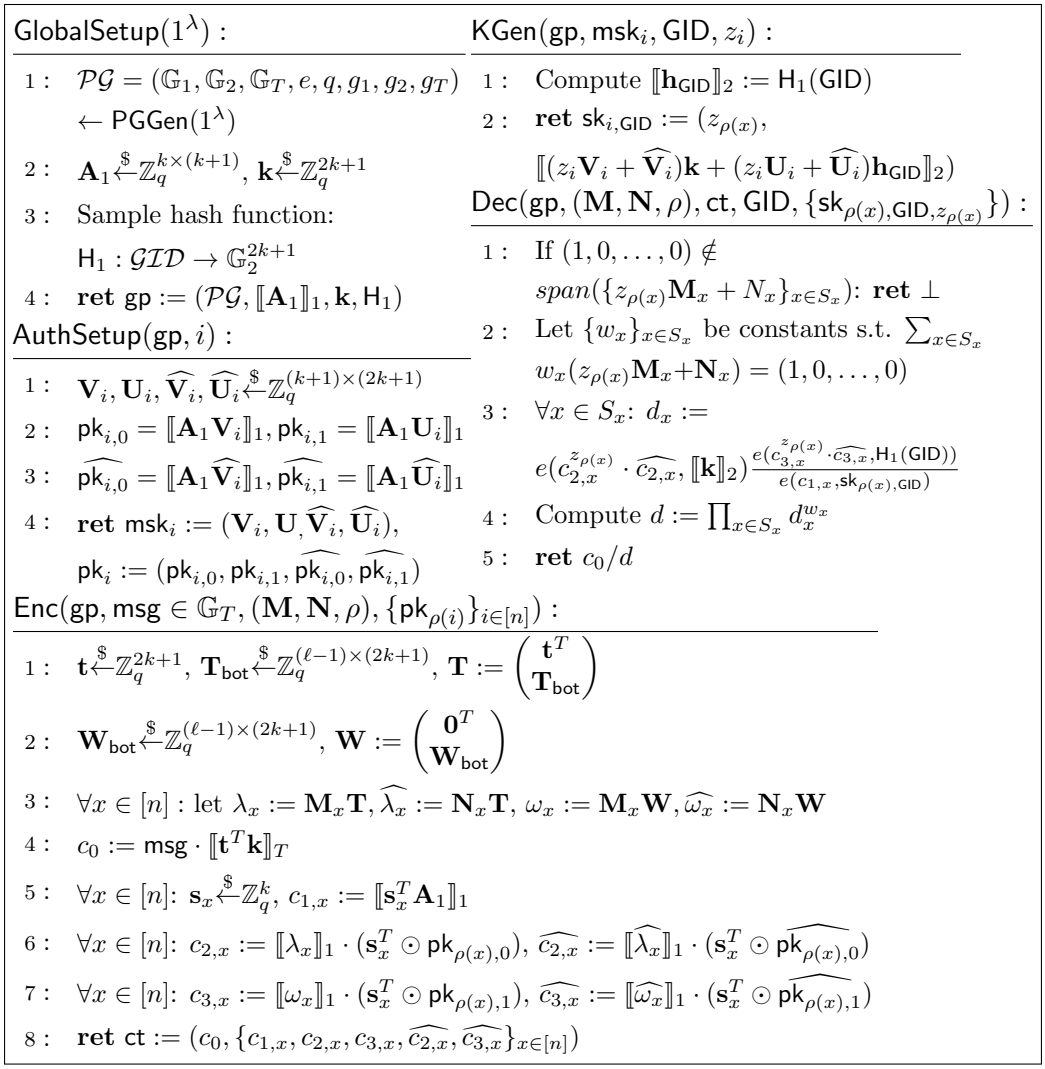

Figure 2: Construction: MA-ABE scheme for ASP from prime-order groups

Theorem 6.1 (Informal)**.** *The MA-ABE construction for ASP in Figure 2 is fully adaptively secure with the additional restrictions that no attribute authority appearing in challenge ciphertext is corrupted and adversary queries at most one key per authority and user pair* (i, GID)*.*

7 Compiler for MA-ABE for ASP: boosting security

We now show how to generically modify our MA-ABE for ASP construction in Section 6 to achieve the security model of Cini et al. [CLW25]. However, it would satisfy the same weak functionality as [CLW25]: decryptors must have keys from all authorities in ciphertext policy in order to decrypt. Nevertheless, unlike [CLW25], our approach does not require a very-selective security model or bounds on number of authorities, and it supports fully adaptive queries, including corruption.

The modified construction is obtained via a compiler consisting of two layers: an inner layer of MA-ABE for ASP with full adaptive security with Type 1 restriction (such as in Figure 2), and an outer layer of MA-ABE for conjunctions with full adaptive security (implied by MA-ABE for

monotone BSP, such as in Figure 1). In the new scheme, to encrypt a message under an ASP policy, we encrypt the message with inner scheme, then encrypt the result under policy "conjunction of all authorities in set U" using outer scheme, where U is set of authorities appearing in the ASP policy.

GlobalSetup, AuthSetup, KGen are run for both layers, with outputs concatenated. Decryption uses all keys from set U to first decrypt outer ciphertext, recovering the inner one, which is then decrypted using the ASP scheme.

The security proof proceeds by guessing at the outset whether the adversary's challenge will avoid corrupted authorities. If yes, we reduce the security to that of inner ASP scheme; if no, to the conjunction scheme. A wrong guess leads to abort, incurring a security loss of $\frac{1}{2}$. Thus, this construction offers a clean way to match [CLW25]'s protection against corrupt authorities while retaining fully adaptivity of our techniques.

Theorem 7.1 (Informal)**.** *There exists an MA-ABE scheme for ASP that is fully adaptive secure with respect to the following restrictions:*

1. *decryption requires keys from all authorities appearing in a ciphertext,*
2. *either the adversary corrupts no authority appearing in the challenge ciphertext policy or for each GID queried, there exists an honest authority appearing in the challenge ciphertext policy who did not issue any secret key,*
3. *the adversary queries at most one key per authority and user id pair* (i, GID)*.*

We provide the formal construction and proof sketch in Appendix E.

Acknowledgements

This research was sponsored by the National Science Foundation under award numbers CNS-2044679 and CNS-2212746. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution, the U.S. government or any other entity.

References

[ABGW17] Miguel Ambrona, Gilles Barthe, Romain Gay, and Hoeteck Wee. Attribute-based encryption in the generic group model: Automated proofs and new constructions. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 2017*, pages 647–664. ACM Press, October / November 2017.

[AFV11] Shweta Agrawal, David Mandell Freeman, and Vinod Vaikuntanathan. Functional encryption for inner product predicates from learning with errors. In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 21–40. Springer, Berlin, Heidelberg, December 2011.

[AG21] Miguel Ambrona and Romain Gay. Multi-authority ABE, revisited. Cryptology ePrint Archive, Report 2021/1381, 2021.

[AHY15] Nuttapong Attrapadung, Goichiro Hanaoka, and Shota Yamada. Conversions among several classes of predicate encryption and applications to ABE with various compactness

tradeoffs. In Tetsu Iwata and Jung Hee Cheon, editors, *ASIACRYPT 2015, Part I*, volume 9452 of *LNCS*, pages 575–601. Springer, Berlin, Heidelberg, November / December 2015.

[AIK11] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. How to garble arithmetic circuits. In Rafail Ostrovsky, editor, *52nd FOCS*, pages 120–129. IEEE Computer Society Press, October 2011.

[AKY24] Shweta Agrawal, Simran Kumari, and Shota Yamada. Attribute based encryption for turing machines from lattices. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part III*, volume 14922 of *LNCS*, pages 352–386. Springer, Cham, August 2024.

[AMY19] Shweta Agrawal, Monosij Maitra, and Shota Yamada. Attribute based encryption (and more) for nondeterministic finite automata from LWE. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part II*, volume 11693 of *LNCS*, pages 765–797. Springer, Cham, August 2019.

[AMYY25] Shweta Agrawal, Anuja Modi, Anshu Yadav, and Shota Yamada. Evasive lwe: Attacks, variants & obfustopia. *Cryptology ePrint Archive*, 2025.

[ARYY23] Shweta Agrawal, Mélissa Rossi, Anshu Yadav, and Shota Yamada. Constant input attribute based (and predicate) encryption from evasive and tensor LWE. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part IV*, volume 14084 of *LNCS*, pages 532–564. Springer, Cham, August 2023.

[AT20] Nuttapong Attrapadung and Junichi Tomida. Unbounded dynamic predicate compositions in ABE from standard assumptions. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part III*, volume 12493 of *LNCS*, pages 405–436. Springer, Cham, December 2020.

[Att14] Nuttapong Attrapadung. Dual system encryption via doubly selective security: Framework, fully secure functional encryption for regular languages, and more. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 557–577. Springer, Berlin, Heidelberg, May 2014.

[Att16] Nuttapong Attrapadung. Dual system encryption framework in prime-order groups via computational pair encodings. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part II*, volume 10032 of *LNCS*, pages 591–623. Springer, Berlin, Heidelberg, December 2016.

[Att19] Nuttapong Attrapadung. Unbounded dynamic predicate compositions in attribute-based encryption. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part I*, volume 11476 of *LNCS*, pages 34–67. Springer, Cham, May 2019.

[AY20] Shweta Agrawal and Shota Yamada. Optimal broadcast encryption from pairings and LWE. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part I*, volume 12105 of *LNCS*, pages 13–43. Springer, Cham, May 2020.

[AYY22] Shweta Agrawal, Anshu Yadav, and Shota Yamada. Multi-input attribute based encryption and predicate encryption. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part I*, volume 13507 of *LNCS*, pages 590–621. Springer, Cham, August 2022.

[BB11] Dan Boneh and Xavier Boyen. Efficient selective identity-based encryption without random oracles. *Journal of Cryptology*, 24(4):659–693, October 2011.

[BBS04] Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In Matthew Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 41–55. Springer, Berlin, Heidelberg, August 2004.

[Bei96a] Amos Beimel. Secure schemes for secret sharing and key distribution. *PhD thesis, Israel Institute of Technology, Technion*, 1996.

[Bei96b] Amos Beimel. Secure schemes for secret sharing and key distribution. *PhD thesis, Israel Institute of Technology, Technion*, 1996.

[BF01] Dan Boneh and Matthew K. Franklin. Identity-based encryption from the Weil pairing. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 213–229. Springer, Berlin, Heidelberg, August 2001.

[BGG+14] Dan Boneh, Craig Gentry, Sergey Gorbunov, Shai Halevi, Valeria Nikolaenko, Gil Segev, Vinod Vaikuntanathan, and Dhinakaran Vinayagamurthy. Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 533–556. Springer, Berlin, Heidelberg, May 2014.

[BL88] Josh Cohen Benaloh and Jerry Leichter. Generalized secret sharing and monotone functions. In *Proceedings of the 8th Annual International Cryptology Conference on Advances in Cryptology*, pages 27–35, 1988.

[Boy13] Xavier Boyen. Attribute-based functional encryption on lattices. In Amit Sahai, editor, *TCC 2013*, volume 7785 of *LNCS*, pages 122–142. Springer, Berlin, Heidelberg, March 2013.

[BSW07] John Bethencourt, Amit Sahai, and Brent Waters. Ciphertext-policy attribute-based encryption. In *2007 IEEE Symposium on Security and Privacy*, pages 321–334. IEEE Computer Society Press, May 2007.

[BÜW24] Chris Brzuska, Akin Ünal, and Ivy K. Y. Woo. Evasive LWE assumptions: Definitions, classes, and counterexamples. In Kai-Min Chung and Yu Sasaki, editors, *ASIACRYPT 2024, Part IV*, volume 15487 of *LNCS*, pages 418–449. Springer, Singapore, December 2024.

[BV16] Zvika Brakerski and Vinod Vaikuntanathan. Circuit-ABE from LWE: Unbounded attributes and semi-adaptive security. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part III*, volume 9816 of *LNCS*, pages 363–384. Springer, Berlin, Heidelberg, August 2016.

[BV20] Zvika Brakerski and Vinod Vaikuntanathan. Lattice-inspired broadcast encryption and succinct ciphertext-policy ABE. Cryptology ePrint Archive, Report 2020/191, 2020.

[CC09] Melissa Chase and Sherman S. M. Chow. Improving privacy and security in multi-authority attribute-based encryption. In Ehab Al-Shaer, Somesh Jha, and Angelos D. Keromytis, editors, *ACM CCS 2009*, pages 121–130. ACM Press, November 2009.

[CCG+23] Jie Chen, Qiaohan Chu, Ying Gao, Jianting Ning, and Luping Wang. Improved fully adaptive decentralized MA-ABE for NC1 from MDDH. In Jian Guo and Ron Steinfeld, editors, *ASIACRYPT 2023, Part V*, volume 14442 of *LNCS*, pages 3–32. Springer, Singapore, December 2023.

[CGKW18] Jie Chen, Junqing Gong, Lucas Kowalczyk, and Hoeteck Wee. Unbounded ABE via bilinear entropy expansion, revisited. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part I*, volume 10820 of *LNCS*, pages 503–534. Springer, Cham, April / May 2018.

[CGW15] Jie Chen, Romain Gay, and Hoeteck Wee. Improved dual system ABE in prime-order groups via predicate encodings. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 595–624. Springer, Berlin, Heidelberg, April 2015.

[Cha07] Melissa Chase. Multi-authority attribute based encryption. In Salil P. Vadhan, editor, *TCC 2007*, volume 4392 of *LNCS*, pages 515–534. Springer, Berlin, Heidelberg, February 2007.

[CLW25] Valerio Cini, Russell Lai, and Ivy Woo. Lattice-based multi-authority/client attribute-based encryption for circuits. *IACR Communications in Cryptology*, 1(4):1–67, 2025.

[CS02] Ronald Cramer and Victor Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In Lars R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 45–64. Springer, Berlin, Heidelberg, April / May 2002.

[CW77] JL Carter and MN Wegman. Universal classes of hash functions (extended abstract), stoc'77: Proceedings of the ninth annual acm symposium on theory of computing, 1977.

[CW23] Valerio Cini and Hoeteck Wee. ABE for circuits with poly (λ)-sized keys from LWE. In *64th FOCS*, pages 435–446. IEEE Computer Society Press, November 2023.

[CW24] Valerio Cini and Hoeteck Wee. Unbounded ABE for circuits from LWE, revisited. In Kai-Min Chung and Yu Sasaki, editors, *ASIACRYPT 2024, Part IV*, volume 15487 of *LNCS*, pages 238–267. Springer, Singapore, December 2024.

[CW25] Valerio Cini and Hoeteck Wee. Faster abe for turing machines from circular evasive lwe. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 94–125. Springer, 2025.

[DHM+24] Fangqi Dong, Zihan Hao, Ethan Mook, Hoeteck Wee, and Daniel Wichs. Laconic function evaluation and ABE for RAMs from (ring-)LWE. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part III*, volume 14922 of *LNCS*, pages 107–142. Springer, Cham, August 2024.

[DKW21] Pratish Datta, Ilan Komargodski, and Brent Waters. Decentralized multi-authority ABE for DNFs from LWE. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part I*, volume 12696 of *LNCS*, pages 177–209. Springer, Cham, October 2021.

[DKW23] Pratish Datta, Ilan Komargodski, and Brent Waters. Fully adaptive decentralized multi-authority ABE. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part III*, volume 14006 of *LNCS*, pages 447–478. Springer, Cham, April 2023.

[dlPVA22] Antonio de la Piedra, Marloes Venema, and Greg Alpár. ABE squared: Accurately benchmarking efficiency of attribute-based encryption. *IACR TCHES*, 2022(2):192–239, 2022.

[dlPVA23] Antonio de la Piedra, Marloes Venema, and Greg Alpár. ACABELLA: Automated (crypt)analysis of attribute-based encryption leveraging linear algebra. In Weizhi Meng, Christian Damsgaard Jensen, Cas Cremers, and Engin Kirda, editors, *ACM CCS 2023*, pages 3269–3283. ACM Press, November 2023.

[DOT19] Pratish Datta, Tatsuaki Okamoto, and Katsuyuki Takashima. Efficient attribute-based signatures for unbounded arithmetic branching programs. In Dongdai Lin and Kazue Sako, editors, *PKC 2019, Part I*, volume 11442 of *LNCS*, pages 127–158. Springer, Cham, April 2019.

[EHK+13] Alex Escala, Gottfried Herold, Eike Kiltz, Carla Rafols, and Jorge Luis Villar. An algebraic framework for diffie-hellman assumptions. In *CRYPTO*, 2013.

[ETS18] ETSI. *CYBER; Application of Attribute Based Encryption (ABE) for PII and personal data protection on IoT devices, WLAN, cloud and mobile services - High level requirements 2018*. ETSI TS 103 458, 2018.

[fil] filecoin-project/blstrs. https://github.com/filecoin-project/blstrs.

[Fre10] David Mandell Freeman. Converting pairing-based cryptosystems from composite-order groups to prime-order groups. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 44–61. Springer, Berlin, Heidelberg, May / June 2010.

[Für07] Martin Fürer. Faster integer multiplication. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 57–66, 2007.

[GDCC16] Junqing Gong, Xiaolei Dong, Jie Chen, and Zhenfu Cao. Efficient IBE with tight reduction to standard assumption in the multi-challenge setting. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part II*, volume 10032 of *LNCS*, pages 624–654. Springer, Berlin, Heidelberg, December 2016.

[GGH+13] Sanjam Garg, Craig Gentry, Shai Halevi, Amit Sahai, and Brent Waters. Attribute-based encryption for circuits from multilinear maps. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 479–499. Springer, Berlin, Heidelberg, August 2013.

[GGL24] Rachit Garg, Rishab Goyal, and George Lu. Dynamic collusion functional encryption and multi-authority attribute-based encryption. In Qiang Tang and Vanessa Teague, editors, *PKC 2024, Part II*, volume 14604 of *LNCS*, pages 69–104. Springer, Cham, April 2024.

[GHKW16] Romain Gay, Dennis Hofheinz, Eike Kiltz, and Hoeteck Wee. Tightly CCA-secure encryption without pairings. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part I*, volume 9665 of *LNCS*, pages 1–27. Springer, Berlin, Heidelberg, May 2016.

[GKW17] Rishab Goyal, Venkata Koppula, and Brent Waters. Lockable obfuscation. In Chris Umans, editor, *58th FOCS*, pages 612–621. IEEE Computer Society Press, October 2017.

[GPSW06] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *ACM CCS 2006*, pages 89–98. ACM Press, October / November 2006. Available as Cryptology ePrint Archive Report 2006/309.

[Gui13] Aurore Guillevic. Comparing the pairing efficiency over composite-order and prime-order elliptic curves. In Michael J. Jacobson, Jr., Michael E. Locasto, Payman Mohassel, and Reihaneh Safavi-Naini, editors, *ACNS 13International Conference on Applied Cryptography and Network Security*, volume 7954 of *LNCS*, pages 357–372. Springer, Berlin, Heidelberg, June 2013.

[GVW13] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Attribute-based encryption for circuits. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th ACM STOC*, pages 545–554. ACM Press, June 2013.

[GW20] Junqing Gong and Hoeteck Wee. Adaptively secure ABE for DFA from k-Lin and more. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part III*, volume 12107 of *LNCS*, pages 278–308. Springer, Cham, May 2020.

[GWW19] Junqing Gong, Brent Waters, and Hoeteck Wee. ABE for DFA from k-Lin. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part II*, volume 11693 of *LNCS*, pages 732–764. Springer, Cham, August 2019.

[HJL25] Yao-Ching Hsieh, Aayush Jain, and Huijia Lin. Lattice-based post-quantum io from circular security with random opening assumption (part ii: zeroizing attacks against private-coin evasive lwe assumptions). *Cryptology ePrint Archive*, 2025.

[HLL23] Yao-Ching Hsieh, Huijia Lin, and Ji Luo. Attribute-based encryption for circuits of unbounded depth from lattices. In *64th FOCS*, pages 415–434. IEEE Computer Society Press, November 2023.

[HLL24] Yao-Ching Hsieh, Huijia Lin, and Ji Luo. A general framework for lattice-based ABE using evasive inner-product functional encryption. In Marc Joye and Gregor Leander, editors, *EUROCRYPT 2024, Part II*, volume 14652 of *LNCS*, pages 433–464. Springer, Cham, May 2024.

[IW14a] Yuval Ishai and Hoeteck Wee. Partial garbling schemes and their applications. In Javier Esparza, Pierre Fraigniaud, Thore Husfeldt, and Elias Koutsoupias, editors, *ICALP 2014, Part I*, volume 8572 of *LNCS*, pages 650–662. Springer, Berlin, Heidelberg, July 2014.

[IW14b] Yuval Ishai and Hoeteck Wee. Partial garbling schemes and their applications. In *International Colloquium on Automata, Languages, and Programming*, pages 650–662. Springer, 2014.

[JKK+17] Zahra Jafargholi, Chethan Kamath, Karen Klein, Ilan Komargodski, Krzysztof Pietrzak, and Daniel Wichs. Be adaptive, avoid overcommitting. In Jonathan Katz and Hovav

Shacham, editors, *CRYPTO 2017, Part I*, volume 10401 of *LNCS*, pages 133–163. Springer, Cham, August 2017.

[JLL23] Aayush Jain, Huijia Lin, and Ji Luo. On the optimal succinctness and efficiency of functional encryption and attribute-based encryption. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part III*, volume 14006 of *LNCS*, pages 479–510. Springer, Cham, April 2023.

[JW16] Zahra Jafargholi and Daniel Wichs. Adaptive security of yao's garbled circuits. Cryptology ePrint Archive, Report 2016/814, 2016.

[KOS16] Marcel Keller, Emmanuela Orsini, and Peter Scholl. MASCOT: Faster malicious arithmetic secure computation with oblivious transfer. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *ACM CCS 2016*, pages 830–842. ACM Press, October 2016.

[KW93] Mauricio Karchmer and Avi Wigderson. On span programs. In *Proceedings of Structures in Complexity Theory*, pages 102–111, 1993.

[KW19] Lucas Kowalczyk and Hoeteck Wee. Compact adaptively secure ABE for NC^1 from k-Lin. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part I*, volume 11476 of *LNCS*, pages 3–33. Springer, Cham, May 2019.

[KW20] Lucas Kowalczyk and Hoeteck Wee. Compact adaptively secure ABE for NC^1 from k-Lin. *Journal of Cryptology*, 33(3):954–1002, July 2020.

[LCLS08] Huang Lin, Zhenfu Cao, Xiaohui Liang, and Jun Shao. Secure threshold multi authority attribute based encryption without a central authority. In Dipanwita Roy Chowdhury, Vincent Rijmen, and Abhijit Das, editors, *INDOCRYPT 2008*, volume 5365 of *LNCS*, pages 426–436. Springer, Berlin, Heidelberg, December 2008.

[LL20a] Huijia Lin and Ji Luo. Compact adaptively secure ABE from k-Lin: Beyond NC^1 and towards NL. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part III*, volume 12107 of *LNCS*, pages 247–277. Springer, Cham, May 2020.

[LL20b] Huijia Lin and Ji Luo. Succinct and adaptively secure ABE for ABP from k-Lin. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part III*, volume 12493 of *LNCS*, pages 437–466. Springer, Cham, December 2020.

[LLL22] Hanjun Li, Huijia Lin, and Ji Luo. ABE for circuits with constant-size secret keys and adaptive security. In Eike Kiltz and Vinod Vaikuntanathan, editors, *TCC 2022, Part I*, volume 13747 of *LNCS*, pages 680–710. Springer, Cham, November 2022.

[LOS+10] Allison B. Lewko, Tatsuaki Okamoto, Amit Sahai, Katsuyuki Takashima, and Brent Waters. Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 62–91. Springer, Berlin, Heidelberg, May / June 2010.

[LVV+23] Watson Ladd, Tanya Verma, Marloes Venema, Armando Faz-Hernández, Brendan McMillion, Avani Wildani, and Nick Sullivan. Portunus: Re-imagining access control in distributed systems. In *2023 USENIX Annual Technical Conference (USENIX ATC 23)*, pages 35–52, 2023.

[LW10] Allison B. Lewko and Brent Waters. New techniques for dual system encryption and fully secure HIBE with short ciphertexts. In Daniele Micciancio, editor, *TCC 2010*, volume 5978 of *LNCS*, pages 455–479. Springer, Berlin, Heidelberg, February 2010.

[LW11a] Allison B. Lewko and Brent Waters. Decentralizing attribute-based encryption. In Kenneth G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 568–588. Springer, Berlin, Heidelberg, May 2011.

[LW11b] Allison B. Lewko and Brent Waters. Unbounded HIBE and attribute-based encryption. In Kenneth G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 547–567. Springer, Berlin, Heidelberg, May 2011.

[LW12] Allison B. Lewko and Brent Waters. New proof methods for attribute-based encryption: Achieving full security through selective techniques. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 180–198. Springer, Berlin, Heidelberg, August 2012.

[MKE09a] Sascha Müller, Stefan Katzenbeisser, and Claudia Eckert. Distributed attribute-based encryption. In Pil Joong Lee and Jung Hee Cheon, editors, *ICISC 08*, volume 5461 of *LNCS*, pages 20–36. Springer, Berlin, Heidelberg, December 2009.

[MKE09b] Sascha Muller, Stefan Katzenbeisser, and Claudia Eckert. On multi-authority ciphertext-policy attribute-based encryption. *Bulletin of the Korean Mathematical Society*, 46(4):803–819, 2009.

[OSW07] Rafail Ostrovsky, Amit Sahai, and Brent Waters. Attribute-based encryption with non-monotonic access structures. In Peng Ning, Sabrina De Capitani di Vimercati, and Paul F. Syverson, editors, *ACM CCS 2007*, pages 195–203. ACM Press, October 2007.

[OT10] Tatsuaki Okamoto and Katsuyuki Takashima. Fully secure functional encryption with general relations from the decisional linear assumption. In Tal Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 191–208. Springer, Berlin, Heidelberg, August 2010.

[OT12] Tatsuaki Okamoto and Katsuyuki Takashima. Fully secure unbounded inner-product and attribute-based encryption. In Xiaoyun Wang and Kazue Sako, editors, *ASIACRYPT 2012*, volume 7658 of *LNCS*, pages 349–366. Springer, Berlin, Heidelberg, December 2012.

[OT20a] Tatsuaki Okamoto and Katsuyuki Takashima. Decentralized attribute-based encryption and signatures. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 103(1):41–73, 2020.

[OT20b] Tatsuaki Okamoto and Katsuyuki Takashima. Decentralized attribute-based encryption and signatures. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 103(1):41–73, 2020.

[PB23] René Peralta and Luís TAN Brandão. Nist first call for multi-party threshold schemes. *(No Title)*, 2023.

[PHGR13] Bryan Parno, Jon Howell, Craig Gentry, and Mariana Raykova. Pinocchio: Nearly practical verifiable computation. In *2013 IEEE Symposium on Security and Privacy*, pages 238–252. IEEE Computer Society Press, May 2013.

[RW15] Yannis Rouselakis and Brent Waters. Efficient statically-secure large-universe multi-authority attribute-based encryption. In Rainer Böhme and Tatsuaki Okamoto, editors, *FC 2015*, volume 8975 of *LNCS*, pages 315–332. Springer, Berlin, Heidelberg, January 2015.

[RW22] Doreen Riepel and Hoeteck Wee. FABEO: Fast attribute-based encryption with optimal security. In Heng Yin, Angelos Stavrou, Cas Cremers, and Elaine Shi, editors, *ACM CCS 2022*, pages 2491–2504. ACM Press, November 2022.

[Sha79] Adi Shamir. How to share a secret. *Communications of the Association for Computing Machinery*, 22(11):612–613, November 1979.

[sup] supranational/blst. https://github.com/supranational/blst.

[SW05] Amit Sahai and Brent R. Waters. Fuzzy identity-based encryption. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 457–473. Springer, Berlin, Heidelberg, May 2005.

[Tom22] Alin Tomescu. Pairings or bilinear maps, 2022. https://alinush.github.io/pairings.

[Tsa19] Rotem Tsabary. Fully secure attribute-based encryption for t-CNF from LWE. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part I*, volume 11692 of *LNCS*, pages 62–85. Springer, Cham, August 2019.

[V+12] Salil P Vadhan et al. Pseudorandomness. *Foundations and Trends® in Theoretical Computer Science*, 7(1–3):1–336, 2012.

[VAH23] Marloes Venema, Greg Alpár, and Jaap-Henk Hoepman. Systematizing core properties of pairing-based attribute-based encryption to uncover remaining challenges in enforcing access control in practice. *DCC*, 91(1):165–220, 2023.

[Ven23] Marloes Venema. A practical compiler for attribute-based encryption: New decentralized constructions and more. In Mike Rosulek, editor, *CT-RSA 2023*, volume 13871 of *LNCS*, pages 132–159. Springer, Cham, April 2023.

[Wat09] Brent Waters. Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 619–636. Springer, Berlin, Heidelberg, August 2009.

[Wat11] Brent Waters. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In Dario Catalano, Nelly Fazio, Rosario Gennaro, and Antonio Nicolosi, editors, *PKC 2011*, volume 6571 of *LNCS*, pages 53–70. Springer, Berlin, Heidelberg, March 2011.

[Wat12] Brent Waters. Functional encryption for regular languages. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 218–235. Springer, Berlin, Heidelberg, August 2012.

[Wee14] Hoeteck Wee. Dual system encryption via predicate encodings. In Yehuda Lindell, editor, *TCC 2014*, volume 8349 of *LNCS*, pages 616–637. Springer, Berlin, Heidelberg, February 2014.

[Wee21a] Hoeteck Wee. ABE for DFA from LWE against bounded collusions, revisited. In Kobbi Nissim and Brent Waters, editors, *TCC 2021, Part II*, volume 13043 of *LNCS*, pages 288–309. Springer, Cham, November 2021.

[Wee21b] Hoeteck Wee. Broadcast encryption with size $N^{1/3}$ and more from k-lin. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part IV*, volume 12828 of *LNCS*, pages 155–178, Virtual Event, August 2021. Springer, Cham.

[Wee22] Hoeteck Wee. Optimal broadcast encryption and CP-ABE from evasive lattice assumptions. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part II*, volume 13276 of *LNCS*, pages 217–241. Springer, Cham, May / June 2022.

[Wee24] Hoeteck Wee. Circuit ABE with $\mathsf{poly}(\mathsf{depth}, \lambda)$-sized ciphertexts and keys from lattices. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part III*, volume 14922 of *LNCS*, pages 178–209. Springer, Cham, August 2024.

[Wee25] Hoeteck Wee. Almost optimal kp and cp-abe for circuits from succinct lwe. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 34–62. Springer, 2025.

[WW24] Brent Waters and Daniel Wichs. Adaptively secure attribute-based encryption from witness encryption. In Elette Boyle and Mohammad Mahmoody, editors, *TCC 2024, Part III*, volume 15366 of *LNCS*, pages 65–90. Springer, Cham, December 2024.

[WWW22] Brent Waters, Hoeteck Wee, and David J. Wu. Multi-authority ABE from lattices without random oracles. In Eike Kiltz and Vinod Vaikuntanathan, editors, *TCC 2022, Part I*, volume 13747 of *LNCS*, pages 651–679. Springer, Cham, November 2022.

A Preliminaries: Appendix

A.1 Access Structures

In this subsection, we present the definition of access structures.

Definition A.1 (Access Structures [BL88, Bei96b]). *Let $\mathbb{U}$ be the attribute universe. An access structure on $\mathbb{U}$ is a collection $\mathbb{A} \subseteq 2^{\mathbb{U}} \setminus \emptyset$ of non-empty sets of attributes. The sets in $\mathbb{A}$ are called the authorized sets and the sets not in $\mathbb{A}$ are called the unauthorized sets. An access structure is called monotone if $\forall B, C \in 2^{\mathbb{U}}$ if $B \in \mathbb{A}$ and $B \subseteq C$, then $C \in \mathbb{A}$.*

A.2 Boolean Span Program

Definition A.2 (Boolean Span Program [KW93, AHY15]). *Let $\mathcal{U} = \{u_1, \ldots, u_m\}$ be a set of variables. For each u_i, denote $\neg u_i$ as a new variable. Intuitively, u_i and $\neg u_i$ correspond to positive and negative attributes, respectively. Also let $\mathcal{U}' = \{\neg u_1, \ldots, \neg u_m\}$. A boolean span program (BSP) over $\mathbb{Z}_q$ is specified by a pair $(\mathbf{M}, \rho)$ of a matrix $\mathbf{M} \in \mathbb{Z}_q^{n \times \ell}$ and a labeling function $\rho : [n] \to \mathcal{U} \cup \mathcal{U}'$ for some integers n and ℓ. Intuitively, the map ρ labels row j with attribute $\rho(j)$.*

A boolean span program accepts or rejects an input by the following criteria. For an input $\mathbf{x} = (x_1, \ldots, x_m) \in \{0,1\}^{1 \times m}$, we define the sub-matrix $\mathbf{M_x}$ of $\mathbf{M}$ to consist of the rows whose labels are set to 1 by the input $\mathbf{x}$. That is, it consists of either rows labelled by some u_i such that $x_i = 1$ or rows labelled by some $\neg u_i$ such that $x_i = 0$. We say that

$$\mathbf{x} \in \{0,1\}^{1 \times m} \text{ satisfies } (\mathbf{M}, \rho) \text{ iff } \mathbf{e}_1 \in \mathsf{rowSpan}(\mathbf{M_x}),$$

where $\mathbf{e}_1 := (1, 0, \ldots, 0) \in \mathbb{Z}_q^{1 \times \ell}$.

Monotone BSP. A BSP is called *monotone* if the labels of the columns consist of only the positive literals, that is, the set $\mathcal{U}$.

A.3 Arithmetic Span Programs

In this section, we recall the definition of arithmetic span programs (ASP) as defined by [IW14b].

Definition A.3 (Arithmetic Span Program [IW14b]). *An arithmetic span program over $\mathbb{Z}_q$ is specified by a tuple $(\mathbf{M}, \mathbf{N}, \rho)$ of matricies $\mathbf{M}, \mathbf{N} \in \mathbb{Z}_q^{n \times \ell}$, and a labeling function $\rho : [n] \to [m]$ for some integers n, ℓ and m. Intuitively, the map ρ labels row j of both matrices $\mathbf{M}$ and $\mathbf{N}$ with $\rho(j)^{th}$ attribute.*

An arithmetic span program accepts or rejects an input by the following criteria. For $j \in [n]$, let $\mathbf{M}_j$ and $\mathbf{N}_j$ denote the j^{th} rows of $\mathbf{M}$ and $\mathbf{N}$ respectively. For an input $\mathbf{x} = (x_1, \ldots, x_m) \in \mathbb{Z}_q^{1 \times m}$, we say that

$$\mathbf{x} \in \mathbb{Z}_q^{1 \times m} \text{ satisfies } (\mathbf{M}, \mathbf{N}, \rho) \text{ iff } \mathbf{e}_1 \in span(\{x_{\rho(j)}\mathbf{M}_j + \mathbf{N}_j\}_{j \in [n]}),$$

where $\mathbf{e}_1 := (1, 0, \ldots, 0) \in \mathbb{Z}_q^{\ell}$ and span refers to linear span of a collection of row vectors.

A.4 Secret Sharing Schemes

In this subsection, we present the definition of secret sharing schemes.

Definition A.4 (Secret Sharing Scheme [Sha79]). *A secret sharing scheme is a tuple of algorithms* $(\mathsf{Share}, \mathsf{Reconstruct})$ *where* Share *is the share algorithm and* $\mathsf{Reconstruct}$ *is the reconstruct algorithm. The share algorithm* Share *takes as input a secret $z \in \mathbb{Z}_q$ and a function $f : \{0,1\}^n \to \{0,1\}$, and outputs a set of shares $\{z_i\}_{i \in [m]}$ together with $\rho : [m] \to \{0, 1, \ldots, n\}$.*

- *Correctness requires that for every* $x \in \{0,1\}^n$*, if* $f(x) = 1$*, then*

$$\mathsf{Reconstruct}(f, x, \{z_i\}_{\rho(i)=0 \cup x_{\rho(i)}=1}) = z.$$

- *Security requires that for every* $x \in \{0,1\}^n$*, if* $f(x) = 0$*, then the shares* $\{z_i\}_{\rho(i)=0 \cup x_{\rho(i)}=1}$ *perfectly hide* z*.*

Definition A.5 (Linear Secret Sharing Scheme (LSSS) [Sha79])**.** *A secret sharing scheme that is said to be linear if* Share *is a linear function of the secret* z *and randomness over* $\mathbb{Z}_q$*, and* Reconstruct *computes a linear function of the shares over* $\mathbb{Z}_q$*, that is,* $z = \sum_{\rho(i)=0 \cup x_{\rho(i)}=1} w_i z_i$ *for some constants* $w_i \in \mathbb{Z}_q$*.*

A.5 MA-ABE

An MA-ABE scheme $\mathsf{MA\text{-}ABE} = (\mathsf{GlobalSetup}, \mathsf{AuthSetup}, \mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ consists of five algorithms whose syntax is given below. Let $\mathcal{AU}$ denote the authority universe. Let $\mathcal{GID}$ denote the universe of global identifiers of the users. Let $\mathbb{M}$ denote the message space. We define MA-ABE for access policies specified by an access structure $\mathbb{A}$ defined on a set $U \subseteq \mathcal{AU}$. We assume that each authority control one attribute, and hence we would use the terms "authority" and "attribute" interchangably. This definition naturally generalizes to the situation in which each authority can potentially control an arbitrary (bounded or unbounded) number of attributes (see [LW11a, RW15]). Our definition below follows [DKW23].

- $\mathsf{gp} \leftarrow \mathsf{GlobalSetup}(1^\lambda)$: The global setup algorithm takes in the security parameter λ in unary representation and outputs the global public parameters gp for the system. We assume that gp includes the descriptions of the universe of attribute authorities $\mathcal{AU}$ and universe of the global identifiers of the users $\mathcal{GID}$. Note that both $\mathcal{AU}$ and $\mathcal{GID}$ are given by $\{0,1\}^\lambda$ in case there is no bound on the number of authorities and users in the system.

- $(\mathsf{pk}_i, \mathsf{msk}_i) \leftarrow \mathsf{AuthSetup}(\mathsf{gp}, i)$: The authority $i \in \mathcal{AU}$ calls the authority setup algorithm during its initialization with the global parameters gp as input and receives back its public and master secret key pair pk_i, msk_i.

- $\mathsf{sk}_{i,\mathsf{GID}} \leftarrow \mathsf{KGen}(\mathsf{gp}, \mathsf{msk}_i, \mathsf{GID})$: The key generation algorithm takes as input the global parameters gp, a master secret key msk_i of an authority $i \in \mathcal{AU}$, and a user's global identifier $\mathsf{GID} \in \mathcal{GID}$. It outputs a secret key $\mathsf{sk}_{i,\mathsf{GID}}$ for the user.

- $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{gp}, \mathsf{msg}, \mathbb{A}, \{\mathsf{pk}_i\}_{i \in U})$: The encryption algorithm takes in the global parameters gp, a message $\mathsf{msg} \in \mathbb{M}$, an access policy $\mathbb{A}$ defined on a set $U \subseteq \mathcal{AU}$, and the set $\{\mathsf{pk}_i\}_{i \in U}$ of public keys for all the authorities in the set U. It outputs a ciphertext ct. We assume that the ciphertext implicitly contains $\mathbb{A}, U$.

- $\mathsf{msg}' \leftarrow \mathsf{Dec}(\mathsf{gp}, \mathsf{ct}, \{\mathsf{sk}_{i,\mathsf{GID}}\}_{i \in S})$: The decryption algorithm takes in the global parameters gp, a ciphertext ct generated with respect to some access policy $\mathbb{A}$, and a collection of keys $\{\mathsf{sk}_{i,\mathsf{GID}}\}_{i \in S}$ corresponding to attribute-user ID pairs $\{(i, \mathsf{GID})\}_{i \in S}$ possessed by a user with global identifier GID. It outputs a message msg' when the collection of attributes associated with the secret keys satisfy the access policy $\mathbb{A}$. Otherwise, decryption fails.

Correctness. An MA-ABE scheme is said to be correct if for every $\lambda \in \mathbb{N}$, $\mathsf{msg} \in \mathbb{M}$, $\mathsf{GID} \in \mathcal{GID}$, every access policy $\mathbb{A}$ defined on a set $U \subseteq \mathcal{AU}$, and every subset of attributes $S \subseteq U$ satisfying the access structure (i.e., $S \in \mathbb{A}$), it holds that

$$\Pr\left[\begin{array}{c} \mathsf{gp} \leftarrow \mathsf{GlobalSetup}(1^\lambda), \\ \forall i \in U : (\mathsf{pk}_i, \mathsf{msk}_i) \leftarrow \mathsf{AuthSetup}(\mathsf{gp}, i) \\ \mathsf{msg}' = \mathsf{msg} : \forall i \in S : \mathsf{sk}_{i,\mathsf{GID}} \leftarrow \mathsf{KGen}(\mathsf{gp}, \mathsf{msk}_i, \mathsf{GID}) \\ \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{gp}, \mathsf{msg}, \mathbb{A}, \{\mathsf{pk}_i\}_{i \in U}) \\ \mathsf{msg}' \leftarrow \mathsf{Dec}(\mathsf{gp}, \mathsf{ct}, \{\mathsf{sk}_{i,\mathsf{GID}}\}_{i \in S}) \end{array}\right] = 1.$$

Fully Adaptive Security. We define the fully adaptive (chosen-plaintext) security for a decentralized MA-ABE scheme, namely, we consider a security game where there could be adaptive secret key queries, adaptive authority corruption queries, and adaptive challenge ciphertext query. This is formalized in the following game between a challenger and an adversary. Note that we will consider two types of authority public keys, those which are honestly generated by the challenger and those which are supplied by the adversary itself where the former type of authority keys can be corrupted by the adversary at any point of time during the game and the latter type of authority keys can potentially be malformed.

The game $\mathsf{MA\text{-}ABE}_{\mathcal{A}}^{\mathsf{fully\text{-}adaptive}}(\lambda)$ consists of the following phases:

Global Setup: The challenger runs $\mathsf{GlobalSetup}$ to generate global public parameters gp and gives it to the adversary.

Query Phase 1: The adversary is allowed to adaptively make a polynomial number of queries of the following types:

- Authority Setup Queries: The adversary request to set up an authority $i \in \mathcal{AU}$ of its choice. If an authority setup query for the same authority i has already been queried before, the challenger aborts. Otherwise, the challenger runs $\mathsf{AuthSetup}$ to create a public/master key pair $(\mathsf{pk}_i, \mathsf{msk}_i)$ for the authority i. The challenger provides pk_i to the adversary and stores $(\mathsf{pk}_i, \mathsf{msk}_i)$. Note that the challenger does not return the generated public/master key pair to the adversary.
- Secret Key Queries: The adversary makes a secret key query by submitting a pair (i, GID) to the challenger, where $\mathsf{GID} \in \mathcal{GID}$ is a global identifier and $i \in \mathcal{AU}$ is an attribute authority. If an authority setup query for the authority i has not been made already, the challenger aborts. Otherwise, the challenger runs KGen using the public/master key pair it already created in response to authority setup query for i and generates a secret key $\mathsf{sk}_{i,\mathsf{GID}}$. The challenger provides $\mathsf{sk}_{i,\mathsf{GID}}$ to the adversary.
- Authority Master Key Queries: The adversary requests the master secret key of an authority $i \in \mathcal{AU}$ to the challenger. If an authority setup query for the authority i has not been made previously, the challenger aborts. Otherwise, the challenger provides the adversary the master secret key msk_i for authority i it created in response to the authority setup query for i.

Challenge Phase: The adversary submits two messages, $\mathsf{msg}_0, \mathsf{msg}_1 \in \mathbb{M}$ and an access policy $\mathbb{A}$ defined on a set $U \subseteq \mathcal{AU}$. The adversary also submits the public keys $\{\mathsf{pk}_i\}$ for a subset of attribute authorities appearing in the access structure $\mathbb{A}$. The authority public keys $\{\mathsf{pk}_i\}$ supplied by the

adversary can potentially be malformed, i.e., can fall outside the range of $\mathsf{AuthSetup}$. The access structure $\mathbb{A}$ and the authority public keys $\{\mathsf{pk}_i\}$ must satisfy the following constraints.

1. Let $U_{\mathcal{A}} \subseteq U$ denote the set of attribute authorities for which the adversary supplied the authority public keys $\{\mathsf{pk}_i\}$. Also let $U_{\mathcal{B}}$ denote the set of attribute authorities for which the challenger created the master public key pairs in response to the authority setup query of the adversary so far. Then, it is required that $U_{\mathcal{A}} \cap U_{\mathcal{B}} = \emptyset$.

2. Let S denote the subset of U containing the authorities in $U_{\mathcal{A}}$ plus the authorities for which the adversary made a master key query so far. For each global identifier $\mathsf{GID} \in \mathcal{GID}$, let S_{GID} denote the subset of U containing authorities i such that the adversary queried a secret key for the pair (i, GID). For each $\mathsf{GID} \in \mathcal{GID}$, it is required that $S \cup S_{\mathsf{GID}} \notin \mathbb{A}$.

The challenger flips a random coin $b \xleftarrow{\$} \{0,1\}$ and generates a ciphertext ct by running the Enc algorithm that encrypts msg_b under the access structure $(\mathbf{M}, \rho)$. The challenger sends ct to the adversary.

Query Phase 2: The adversary is allowed to make all types of queries as in Query Phase 1 as long as they do not violate the constraints Properties 1 and 2 above.

Guess: The adversary must submit a guess $b' \in \{0,1\}$ for b. The adversary wins if $b = b'$.

The game is formally defined in Figure 3. The advantage of an adversary $\mathcal{A}$ in this game is defined as

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{MA\text{-}ABE,fully\text{-}adaptive}}(\lambda) = |\Pr[\mathsf{MA\text{-}ABE}_{\mathcal{A}}^{\mathsf{fully\text{-}adaptive}}(\lambda) = 1] - 1/2|.$$

Definition A.6 (Fully adaptive security of MA-ABE)**.** *An MA-ABE scheme for access structure $\mathbb{A}$ is fully adaptively secure if for any p.p.t. adversary $\mathcal{A}$ there exists a negligible function $\mathsf{negl}(\cdot)$ such that for all $\lambda \in \mathbb{N}$, we have $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{MA\text{-}ABE,fully\text{-}adaptive}}(\lambda) \leq \mathsf{negl}(\lambda)$.*

Remark A.7 (Fully adaptive security of MA-ABE in the Random Oracle Model)**.** *: Similar to [LW11a, RW15, OT20b], we additionally consider the aforementioned notion of fully adaptive security in the random oracle model. In this context, we assume a global hash function H published as part of the global public parameters and accessible by all the parties in the system, including the adversary. In the security proof, we model H as a random function and allow it to be programmed by the challenger. Therefore, in the fully adaptive security game described above, we further let the adversary adaptively submit H-oracle queries to the challenger, along with the key queries it makes both before and after the challenge ciphertext query.*

A.6 Assumptions

Assumption A.8 (Matrix Decisional Diffie Hellman Assumption: $\mathsf{MDDH}_{k,\ell}^{\mathbb{G}_i}$ [EHK+13])**.** *Let $\ell > k > 1$. We say that the $\mathsf{MDDH}_{k\ell}^{\mathbb{G}_i}$ assumption holds with respect to PGGen if for all p.p.t. adversary $\mathcal{A}$ and for all $i \in \{1, 2, T\}$, the following advantage is negligible in λ.*

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{MDDH}_{k,\ell}^{\mathbb{G}_i}}(\lambda) := |\Pr[\mathcal{A}(\mathcal{D}, [\![\mathbf{t}_0]\!]_i) = 1] - \Pr[\mathcal{A}(\mathcal{D}, [\![\mathbf{t}_1]\!]_i) = 1]|$$

where $\mathcal{PG} := (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, q, g_1, g_2, g_T) \leftarrow \mathsf{PGGen}(1^\lambda)$, $\mathcal{D} = (\mathcal{PG}, [\![\mathbf{X}]\!]_i), \mathbf{t}_0 = \mathbf{Xu}, \mathbf{t}_1 \xleftarrow{\$} \mathbb{Z}_q^\ell$ such that $\mathbf{X} \xleftarrow{\$} \mathbb{Z}_q^{\ell \times k}$, $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^k$.

Game $\mathsf{MA\text{-}ABE}_{\mathcal{A}}^{\mathsf{fully\text{-}adaptive}}(\lambda)$

1 : $\mathsf{gp} \leftarrow \mathsf{GlobalSetup}(1^\lambda)$
2 : $U_{\mathcal{B}} := \emptyset, U_{\mathcal{C}} := \emptyset$
3 : $\mathsf{SampledKeyMap} := \emptyset, \mathsf{GAMap} := \emptyset$
4 : $\mathsf{chalFlag} := \mathsf{false}$
5 : $(\mathsf{msg}_0, \mathsf{msg}_1, \mathbb{A}, U, \{\mathsf{pk}_i\}_{i \in U_{\mathcal{A}}}) \leftarrow \mathcal{A}^{\mathsf{AuthSetup}(\cdot), \mathsf{KGen}(\cdot,\cdot), \mathsf{Corrupt}(\cdot)}(\mathsf{gp})$
6 : If $\neg\mathsf{Admissible}(U_{\mathcal{B}}, U_{\mathcal{C}}, \mathsf{GAMap})$: **ret** 0
7 : $b \xleftarrow{\$} \{0,1\}$
8 : $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{gp}, \mathsf{msg}_b, \mathbb{A}, \{\mathsf{pk}_i\}_{i \in U_{\mathcal{A}} \cup U_{\mathcal{B}}})$
9 : $\mathsf{chalFlag} := \mathsf{true}$
10 : $b' \leftarrow \mathcal{A}^{\mathsf{AuthSetup}(\cdot), \mathsf{KGen}(\cdot,\cdot), \mathsf{Corrupt}(\cdot)}(\mathsf{ct})$
11 : **ret** 1 if $b = b'$, else 0

Function $\mathsf{Admissible}(U_{\mathcal{B}}, U_{\mathcal{C}}, \mathsf{GAMap})$:

1 : If $U_{\mathcal{A}} \cap U_{\mathcal{B}} \neq \emptyset$: **ret** false
2 : Let $S := U_{\mathcal{A}} \cup U_{\mathcal{C}}$
3 : $\forall \mathsf{GID} \in \mathsf{GAMap}$: let $S_{\mathsf{GID}} := U \cap \mathsf{GAMap}[\mathsf{GID}]$
4 : If $\exists \mathsf{GID} \in \mathsf{GAMap}$ s.t. $S \cup S_{\mathsf{GID}} \in \mathbb{A}$: **ret** false
5 : **ret** true

Oracle $\mathsf{AuthSetup}(i)$:

1 : If $i \in U_{\mathcal{B}}$: **ret** $\perp$
2 : $U_{\mathcal{B}} := U_{\mathcal{B}} \cup \{i\}$
3 : If $\mathsf{chalFlag} \cap \neg\mathsf{Admissible}(U_{\mathcal{B}}, U_{\mathcal{C}}, \mathsf{GAMap})$: **ret** $\perp$
4 : $(\mathsf{pk}_i, \mathsf{msk}_i) \leftarrow \mathsf{AuthSetup}(\mathsf{gp}, i)$
5 : $\mathsf{SampledKeyMap}[i] := (\mathsf{pk}_i, \mathsf{msk}_i)$
6 : **ret** pk_i

Oracle $\mathsf{KGen}(i, \mathsf{GID})$:

1 : If $i \notin U_{\mathcal{B}}$: **ret** $\perp$
2 : $\mathsf{GAMap}[\mathsf{GID}] := \mathsf{GAMap}[\mathsf{GID}] \cup \{i\}$
3 : If $\mathsf{chalFlag} \cap \neg\mathsf{Admissible}(U_{\mathcal{B}}, U_{\mathcal{C}}, \mathsf{GAMap})$: **ret** $\perp$
4 : $(\cdot, \mathsf{msk}_i) = \mathsf{SampledKeyMap}[i]$
5 : $\mathsf{sk}_{i,\mathsf{GID}} \leftarrow \mathsf{KGen}(\mathsf{gp}, \mathsf{msk}_i, \mathsf{GID})$
6 : **ret** $\mathsf{sk}_{i,\mathsf{GID}}$

Oracle $\mathsf{Corrupt}(i)$:

1 : If $i \notin U_{\mathcal{B}}$: **ret** $\perp$
2 : $U_{\mathcal{C}} := U_{\mathcal{C}} \cup \{i\}$
3 : If $\mathsf{chalFlag} \cap \neg\mathsf{Admissible}(U_{\mathcal{B}}, U_{\mathcal{C}}, \mathsf{GAMap})$: **ret** $\perp$
4 : $(\cdot, \mathsf{msk}_i) = \mathsf{SampledKeyMap}[i]$
5 : **ret** msk_i

Figure 3: Security game for MA-ABE.

When $\ell = k + 1$, we simply call it k-MDDH assumption over group $\mathbb{G}_i$.

Next, we describe some lemmas that are prime-order analogues of Subgroup Decision assumptions over composite-order bilinear groups. We describe these with respect to following set of matrices: Fix parameters $\ell_1, \ell_2, \ell_3, \ell_W$. Pick random

$$\mathbf{B}_1 \xleftarrow{\$} \mathbb{Z}_q^{\ell \times \ell_1}, \mathbf{B}_2 \xleftarrow{\$} \mathbb{Z}_q^{\ell \times \ell_2}, \mathbf{B}_3 \xleftarrow{\$} \mathbb{Z}_q^{\ell \times \ell_3},$$

where $\ell := \ell_1 + \ell_2 + \ell_3$. Let $(\mathbf{B}_1^*, \mathbf{B}_2^*, \mathbf{B}_3^*)^T = (\mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3)^{-1}$ so that $\mathbf{B}_i^T \mathbf{B}_i^* = \mathbf{I}$ (known as *non-degeneracy*) and $\mathbf{B}_i^T \mathbf{B}_j^* = \mathbf{0}$ if $i \neq j$ (known as *orthogonality*).

Assumption A.9 (Subgroup Decision Assumption $\mathsf{SD}^{\mathbb{G}_2}_{\mathbf{B}_i \to \mathbf{B}_i, \mathbf{B}_j}$ for $i, j \in \{1,2,3\}$ [CGKW18, GHKW16, GDCC16])**.** *For all $i, j \in \{1,2,3\}$ such that $i \neq j$, the $\mathsf{SD}^{\mathbb{G}_2}_{\mathbf{B}_i \to \mathbf{B}_i, \mathbf{B}_j}$ assumptions states that for any p.p.t. adversary $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\cdot)$ such that for any security parameter $\lambda \in \mathbb{N}$ and for all $k \in \{1,2,3\} \setminus \{i, j\}$,*

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{SD}^{\mathbb{G}_2}_{\mathbf{B}_i \to \mathbf{B}_i, \mathbf{B}_j}}(\lambda) := |\Pr[\mathcal{A}(\mathcal{D}, [\![\mathbf{t}_0]\!]_2) = 1] - \Pr[\mathcal{A}(\mathcal{D}, [\![\mathbf{t}_1]\!]_2) = 1]| \leq \mathsf{negl}(\lambda)$$

where

$$\mathcal{PG} := (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, q, g_1, g_2, g_T) \leftarrow \mathsf{PGGen}(1^\lambda),$$
$$\mathcal{D} = (\mathcal{PG}, [\![\mathbf{B}_1]\!]_2, [\![\mathbf{B}_2]\!]_2, [\![\mathbf{B}_3]\!]_2, \mathsf{basis}(\mathbf{B}_i^*), \mathsf{basis}(\mathbf{B}_k^*), \mathsf{basis}(\mathbf{B}_i^*, \mathbf{B}_j^*)),$$
$$\mathbf{t}_0 \leftarrow \mathsf{span}(\mathbf{B}_i), \mathbf{t}_1 \leftarrow \mathsf{span}(\mathbf{B}_i, \mathbf{B}_j).$$

B MA-ABE for monotone BSP from prime-order groups: Appendix

B.1 Correctness of MA-ABE for monotone BSP construction

For the MA-ABE for monotone BSP construction in Figure 1, observe that d_x can be simplified as follows:

$$\begin{aligned} d_x &= e(c_{2,x}, [\![\mathbf{k}]\!]_2) \cdot \frac{e(c_{3,x}, \mathsf{H}_1(\mathsf{GID}))}{e(c_{1,x}, \mathsf{sk}_{\rho(x),\mathsf{GID}})} \\ &= [\![(\lambda_x + \mathbf{s}_x^T \mathbf{A}\mathbf{V}_{\rho(x)})\mathbf{k} + (\omega_x + \mathbf{s}_x^T \mathbf{A}\mathbf{U}_{\rho(x)})\mathbf{h}_{\mathsf{GID}} - (\mathbf{s}_x^T \mathbf{A})(\mathbf{V}_{\rho(x)}\mathbf{k} + \mathbf{U}_{\rho(x)}\mathbf{h}_{\mathsf{GID}})]\!]_T \\ &= [\![\lambda_x \mathbf{k} + \omega_x \mathbf{h}_{\mathsf{GID}}]\!]_T \\ &= [\![\mathbf{M}_x(\mathbf{T}\mathbf{k} + \mathbf{W}\mathbf{h}_{\mathsf{GID}})]\!]_T. \end{aligned}$$

Then, d can be simplified as follows:

$$\begin{aligned} d &= \prod_{\rho(x)\in S} d_x^{w_x} \\ &= [\![\sum_{\rho(x)\in S} w_x \mathbf{M}_x(\mathbf{T}\mathbf{k} + \mathbf{W}\mathbf{h}_{\mathsf{GID}})]\!]_T \\ &= [\![(1, 0, \ldots, 0)(\mathbf{T}\mathbf{k} + \mathbf{W}\mathbf{h}_{\mathsf{GID}})]\!]_T \\ &= [\![\mathbf{t}^T\mathbf{k} + \mathbf{0}^T\mathbf{h}_{\mathsf{GID}}]\!]_T \qquad \text{(because first rows of } \mathbf{T} \text{ and } \mathbf{W} \text{ are } \mathbf{t}^T \text{ and } \mathbf{0}^T) \\ &= [\![\mathbf{t}^T\mathbf{k}]\!]_T. \end{aligned}$$

Thus, it follows that $c_0/d = \mathsf{msg}$. Thus, correctness holds.

B.2 Missing Proofs from Section 4.1

Before we present the missing proofs, we first define the Q-fold MDDH assumption and comment on the random self-reducibility of the MDDH assumption since these will help simplify the proofs. The Q-fold MDDH assumption is considers Q many independent instances of the MDDH assumption. While using a standard hybrid argument, it can be shown that this is equivalent to the MDDH assumption with a loss of Q in the reduction, but random self-reducibility allows us to give a tighter bound when $Q > \ell - k$ that we state below.

Assumption B.1 (Q-fold MDDH Assumption: $\mathsf{MDDH}^{\mathbb{G}_i}_{k,\ell,Q}$ [EHK+13])**.** *We say that the* $\mathsf{MDDH}^{\mathbb{G}_i}_{k_\ell}$ *assumption holds with respect to* PGGen *if for all p.p.t. adversary* $\mathcal{A}$ *and for all* $i \in \{1, 2, T\}$*, the following advantage is negligible in* λ*.*

$$\mathsf{Adv}^{\mathsf{MDDH}^{\mathbb{G}_i}_{k,\ell,Q}}_{\mathcal{A}}(\lambda) := |\Pr[\mathcal{A}(\mathcal{D}, [\![\mathbf{T}_0]\!]_i) = 1] - \Pr[\mathcal{A}(\mathcal{D}, [\![\mathbf{T}_1]\!]_i) = 1]|$$

where

$$\mathcal{PG} := (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, q, g_1, g_2, g_T) \leftarrow \mathsf{PGGen}(1^\lambda),$$
$$\mathcal{D} = (\mathcal{PG}, [\![\mathbf{X}]\!]_i),$$
$$\mathbf{T}_0 = \mathbf{XU}, \mathbf{T}_1 \xleftarrow{\$} \mathbb{Z}_q^{\ell \times Q}$$

such that $\mathbf{X} \xleftarrow{\$} \mathbb{Z}_q^{\ell \times k}$, $\mathbf{U} \xleftarrow{\$} \mathbb{Z}_p^{k \times Q}$.

Lemma B.2 (Random Self-reducibility of MDDH [EHK+13])**.** *For all* $i \in \{1, 2, T\}$, $\mathsf{MDDH}_{k,\ell}^{\mathbb{G}_i}$ *assumption is random self-reducible. Concretely, for any* Q,

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{MDDH}_{k,\ell,Q}^{\mathbb{G}_i}}(\lambda) \leq \begin{cases} Q \cdot \mathsf{Adv}_{\mathcal{A}}^{\mathsf{MDDH}_{k,\ell}^{\mathbb{G}_i}}(\lambda) & , \text{ if } 1 \leq Q \leq \ell - k \\ (\ell - k) \cdot \mathsf{Adv}_{\mathcal{A}}^{\mathsf{MDDH}_{k,\ell}^{\mathbb{G}_i}}(\lambda) + \frac{1}{q-1} & , \text{ if } Q > \ell - k \end{cases},$$

and the probability is taken over $\mathcal{PG} := (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, q, g_1, g_2, g_T) \leftarrow \mathsf{PGGen}(1^\lambda)$, $\mathbf{X} \xleftarrow{\$} \mathbb{Z}_q^{\ell \times k}$, $\mathbf{U} \xleftarrow{\$} \mathbb{Z}_p^{k \times Q}$, $\mathbf{T}_1 \xleftarrow{\$} \mathbb{Z}_q^{\ell \times Q}$ *and the random coins of* $\mathcal{A}$.

Proof of Claim 4.2. $\mathsf{Hyb}_{\mathsf{Real}}$ and $\mathsf{Hyb}'_{\mathsf{Real}}$ are identically distributed because the distribution of gp in $\mathsf{GlobalSetup}$ and $\mathsf{GlobalSetup}^*$ is the same. □

Proof of Claim 4.3. We show that if there exists a PPT adversary $\mathcal{A}$ that can distinguish between $\mathsf{Hyb}'_{\mathsf{Real}}$ and Hyb_0, then we can use $\mathcal{A}$ to construct a PPT adversary $\mathcal{B}$ that can break the $\mathsf{MDDH}_{k,2k+1,Q}^{\mathbb{G}_2}$ assumption in $\mathbb{G}_2$, where Q is the number of queries made by $\mathcal{A}$ to the H_1 oracle.

Let $\mathcal{B}$ be an adversary that is given a $\mathsf{MDDH}_{k,2k+1,Q}^{\mathbb{G}_2}$ challenge by the challenger $\mathcal{C}$: $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, q, g_1, g_2, g_T, [\![\mathbf{B}_1]\!]_2, [\![\mathbf{H}]\!]_2)$ where $\mathbf{B}_1 \xleftarrow{\$} \mathbb{Z}_q^{(2k+1)\times k}$ and $\mathbf{H} \in \mathbb{Z}_q^{(2k+1)\times Q}$. Here $\mathcal{C}$ samples $b \xleftarrow{\$} \{0,1\}$ and sets $\mathbf{H} = \mathbf{B}_1\mathbf{Z}$ if $b = 0$ and $\mathbf{H} \xleftarrow{\$} \mathbb{Z}_q^{(2k+1)\times Q}$ if $b = 1$. The goal of $\mathcal{B}$ is to output a bit b' and $\mathcal{B}$ wins if $b' = b$.

$\mathcal{B}$ will internally run $\mathcal{A}$ as follows. For simulating the view to $\mathcal{A}$, $\mathcal{B}$ will run exactly the same as $\mathsf{Hyb}'_{\mathsf{Real}}$, except the following modifications:

- $\mathcal{B}$ will use $[\![\mathbf{B}_1]\!]_2$ obtained from the MDDH challenger $\mathcal{C}$ instead of computing it as in $\mathsf{GlobalSetup}^*$ in $\mathsf{Hyb}'_{\mathsf{Real}}$. While $\mathcal{B}$ can sample $\mathbf{B}_2$ and $\mathbf{B}_3$ on its own as in $\mathsf{Hyb}'_{\mathsf{Real}}$, it will not be able to compute $\mathbf{B}_1^*, \mathbf{B}_2^*, \mathbf{B}_3^*$ as in $\mathsf{GlobalSetup}^*$ since it does not know the value of $\mathbf{B}_1$ in the clear. But this is okay since these values are not needed to simulate the view of $\mathcal{A}$.
- $\mathcal{B}$ will embed the challenge $[\![\mathbf{H}]\!]_2$ in the H_1 oracle queries from $\mathcal{A}$ as follows. $\mathcal{B}$ parses $([\![\mathbf{h}_1]\!]_2, \ldots, [\![\mathbf{h}_Q]\!]_2) = [\![\mathbf{H}]\!]_2$, where $[\![\mathbf{h}_j]\!] \in \mathbb{G}_2^{2k+1}$ for all $j \in [Q]$. For the j^{th} query to the H_1 oracle, $\mathcal{B}$ programs $\mathsf{H}_1(\mathsf{GID}_j) := [\![\mathbf{h}_j]\!]_2$.

Observe that when $\mathcal{C}$ chooses $b = 0$, that is, $\mathbf{H} = \mathbf{B}_1\mathbf{Z}$, then $\mathcal{B}$ is able to simulate the view of $\mathcal{A}$ in Hyb_0. Similarly, when $\mathcal{C}$ chooses $b = 1$, that is, $\mathbf{H} \xleftarrow{\$} \mathbb{Z}_q^{(2k+1)\times Q}$, then $\mathcal{B}$ is able to simulate the view of $\mathcal{A}$ in $\mathsf{Hyb}'_{\mathsf{Real}}$. Thus, if $\mathcal{A}$ can distinguish between $\mathsf{Hyb}'_{\mathsf{Real}}$ and Hyb_0, then $\mathcal{B}$ can distinguish between the two cases of the MDDH challenge.

□

Proof of Claim 4.4. Recall that the only difference between Hyb_0 and Hyb_1 is in how the challenge ciphertext is generated. Specifically, for all rows $x \in \overline{U_{\mathcal{A}}}$, the two hybrids differ in how the vector $\mathbf{c}_x \in \mathbb{Z}_q^{k+1}$ is computed. In Hyb_0, $\mathbf{c}_x^T := \mathbf{s}_x^T\mathbf{A}_1$, or equivalently, $\mathbf{c}_x = \mathbf{A}_1^T\mathbf{s}_x$ where $\mathbf{s}_x \xleftarrow{\$} \mathbb{Z}_q^k$ is a random vector. In Hyb_1, $\mathbf{c}_x \xleftarrow{\$} \mathbb{Z}_q^{k+1}$ is a random vector.

We show that if there exists a PPT adversary $\mathcal{A}$ that can distinguish between Hyb_0 and Hyb_1, then we can use $\mathcal{A}$ to construct a PPT adversary $\mathcal{B}$ that can break the $\mathsf{MDDH}^{\mathbb{G}_1}_{k,k+1,|\overline{U_\mathcal{A}}|}$ assumption in $\mathbb{G}_1$, where $\overline{U_\mathcal{A}}$ denotes the subset of rows of challenge access policy matrix $\mathbf{M}$ that are honestly generated by the challenger.

Let $\mathcal{B}$ be an adversary that is given a $\mathsf{MDDH}^{\mathbb{G}_1}_{k,k+1,|\overline{U_\mathcal{A}}|}$ challenge by the challenger $\mathcal{C}$: $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, q, g_1, g_2, g_T, [\![\mathbf{A}_1^T]\!]_1, [\![\mathbf{C}]\!]_1)$ where $\mathbf{A}_1^T \xleftarrow{\$} \mathbb{Z}_q^{(k+1)\times k}$ and $\mathbf{C} \in \mathbb{Z}_q^{(k+1)\times|\overline{U_\mathcal{A}}|}$. Here $\mathcal{C}$ samples $b \xleftarrow{\$} \{0,1\}$ and sets $\mathbf{C} = \mathbf{A}_1^T\mathbf{Z}$ if $b = 0$ and $\mathbf{C} \xleftarrow{\$} \mathbb{Z}_q^{(k+1)\times|\overline{U_\mathcal{A}}|}$ if $b = 1$. The goal of $\mathcal{B}$ is to output a bit b' and $\mathcal{B}$ wins if $b' = b$. $\mathcal{B}$ will internally run $\mathcal{A}$ as follows. For simulating the view to $\mathcal{A}$, $\mathcal{B}$ will run exactly the same as Hyb_0, except the following modifications:

- $\mathcal{B}$ will use $[\![\mathbf{A}_1]\!]_1$ obtained from the MDDH challenger $\mathcal{C}$ instead of computing it as in $\mathsf{GlobalSetup}^*$ in Hyb_0. While $\mathcal{B}$ can sample $\mathbf{A}_2$ on its own as in Hyb_0, it will not be able to compute $\mathbf{A}_1^*, \mathbf{A}_2^*$ as in $\mathsf{GlobalSetup}^*$ since it does not know the value of $\mathbf{A}_1$ in the clear. But this is okay since these values are not needed to simulate the view of $\mathcal{A}$.
- $\mathcal{B}$ will embed the challenge $[\![\mathbf{C}]\!]_1$ in the challege ciphertext query $(\mathsf{msg}_0, \mathsf{msg}_1, (\mathbf{M}, \rho), \{\mathsf{pk}_x\}_{x\in U_\mathcal{A}})$ as follows: $\mathcal{B}$ parses $([\![\mathbf{c}_1]\!]_1, \ldots, [\![\mathbf{c}_{|\overline{U_\mathcal{A}}|}]\!]_1) = [\![\mathbf{C}]\!]_1$, where $[\![\mathbf{c}_x]\!]_1 \in \mathbb{G}_1^{k+1}$ for all $x \in [|\overline{U_\mathcal{A}}|]$. Without loss of generality suppose that $\overline{U_\mathcal{A}} = \{1, 2, \ldots, |\overline{U_\mathcal{A}}|\}$. Then, for all $x \in \overline{U_\mathcal{A}}$, $\mathcal{B}$ uses the challenge $[\![\mathbf{c}_x]\!]_1$ to compute the challenge ciphertext components $c_{1,x}, c_{2,x}, c_{3,x}$ as follows:

$$
\begin{aligned}
c_{1,x} &:= [\![\mathbf{c}_x^T]\!]_1, \\
c_{2,x} &:= [\![\lambda_x]\!]_1 \cdot ([\![\mathbf{c}_x^T]\!]_1 \odot \mathbf{V}_{\rho(x)}), \\
c_{3,x} &:= [\![\omega_x]\!]_1 \cdot ([\![\mathbf{c}_x^T]\!]_1 \odot \mathbf{U}_{\rho(x)}).
\end{aligned}
$$

Observe that when $\mathcal{C}$ chooses $b = 0$, that is, $\mathbf{C} = \mathbf{A}_1^T\mathbf{Z}$, then $\mathcal{B}$ is able to simulate the view of $\mathcal{A}$ in Hyb_0. Similarly, when $\mathcal{C}$ chooses $b = 1$, that is, $\mathbf{C} \xleftarrow{\$} \mathbb{Z}_q^{(2k+1)\times Q}$, then $\mathcal{B}$ is able to simulate the view of $\mathcal{A}$ in Hyb_1. Thus, if $\mathcal{A}$ can distinguish between Hyb_0 and Hyb_1, then $\mathcal{B}$ can distinguish between the two cases of the MDDH challenge.

□

Proof of Claim 4.5. Observe that the only difference between Hyb_1 and Hyb_2 is that in ciphertext component $c_{3,x}$ for all $x \in [n]$: $c_{3,x}$ contains $[\![\omega_x]\!]_1$, where ω_x is a secret share of $\mathbf{0}^T \in \mathbb{Z}_q^{1\times(2k+1)}$ in Hyb_1, but it is a secret share of $\gamma\mathbf{B}_3^{*T}$ in Hyb_2. Therefore, to prove that the hybrids are statistically indistinguishable, we will argue that $\gamma\mathbf{B}_3^{*T}$ is information theoretically hidden from the adversary $\mathcal{A}$ in Hyb_2.

Suppose the challenge access policy $(\mathbf{M}, \rho)$ is defined over a set of authorities $U \subseteq \mathcal{AU}$, that is, $\rho : [n] \to U$. Recall from Appendix A.5 that the game condition requires that $U_\mathcal{A} \cap U_\mathcal{B} = \emptyset$ and for each $\mathsf{GID} \in \mathcal{GID}$, it is required that $S \cup S_{\mathsf{GID}} \notin (\mathbf{M}, \rho)$, where

- $U_\mathcal{A} \subseteq U$ denotes the set of attribute authorities for which the adversary supplied the authority public keys $\{\mathsf{pk}_i\}$,
- $U_\mathcal{B}$ denote the set of attribute authorities for which the challenger created the master public key pairs in response to the authority setup query of the adversary so far,
- S denotes the subset of U containing the authorities in $U_\mathcal{A}$ plus the authorities for which the adversary made a master key query so far (in other words, S denotes the set of corrupt authorities),

- for each global identifier $\mathsf{GID} \in \mathcal{GID}$, S_{GID} denotes the subset of U containing authorities i such that the adversary queried a secret key for the pair (i, GID).

To show that $\gamma \mathbf{B}_3^{*T}$ is information theoretically hidden from the adversary $\mathcal{A}$ in Hyb_2, we only need to rely on $S \notin (\mathbf{M}, \rho)$ which is implied by the second game condition. Here, $S \notin (\mathbf{M}, \rho)$ is a shorthand for $(1, 0, \ldots, 0) \notin \mathsf{rowSpan}(\{\mathbf{M}_x\}_{\rho(x) \in S})$.

Note that the vectors $\mathbf{M}_x \mathbf{W}$ for all rows x of the challenge access matrix $\mathbf{M}$ labeled by corrupt authorities (that is, $\rho(x) \in S$) are information theoretically revealed to $\mathcal{A}$. However, by the game condition the subspace spanned by those rows does not include the vector $(1, 0, \ldots, 0)$. This means that there must exist some vector $\mathbf{u} \in \mathbb{Z}_q^\ell$ such that $\mathbf{u}$ is orthogonal to all these rows of $\mathbf{M}$ (that is, $\mathbf{M}_x \mathbf{u} = 0$) but is not orthogonal to $(1, 0, \ldots, 0)$, that is, the first entry of $\mathbf{u}$ must be non-zero. We consider a basis $\mathbb{U}$ of $\mathbb{Z}_q^\ell$ involving the vector $\mathbf{u}$ and write $\mathbf{W} = \begin{pmatrix} \gamma \mathbf{B}_3^{*T} \\ \mathbf{W}_{\mathsf{bot}} \end{pmatrix} = \widetilde{\mathbf{U}} + \mathbf{u}\mathbf{b}^T$ for some $\mathbf{b} \in \mathbb{Z}_q^{2k+1}$ and some $\widetilde{\mathbf{U}} \in \mathbb{Z}_q^{\ell \times (2k+1)}$ such that each column of $\widetilde{\mathbf{U}}$ lies in the column span of $\mathbb{U} \setminus \mathbf{u}$. Hence, $\widetilde{\mathbf{U}}$ reveals no information about $\mathbf{b}$. Now since the first entry of $\mathbf{u}$ is non-zero, it follows that the first row of $\mathbf{W}$, that is, $\gamma \mathbf{B}_3^{*T}$, depends on $\mathbf{b}$. But $\mathbf{M}_x \mathbf{W}$ for all the corrupted rows of $\mathbf{M}$ contains no information about $\mathbf{b}$ since $\mathbf{u}$ is orthogonal to all these rows. Thus, it follows that these rows do not leak information of $\gamma \mathbf{B}_3^{*T}$.

Therefore, the only possible way for $\mathcal{A}$ to get information about $\gamma \mathbf{B}_3^{*T}$ is through the ciphertext components $c_{3,x}$ corresponding to the uncorrupted rows of $\mathbf{M}$. However, for each such row x, $\mathcal{A}$ can only recover $\mathbf{c}_x^T$, $\mathbf{M}_x \mathbf{W} + \mathbf{c}_x^T \mathbf{U}_{\rho(x)}$ information theoretically. Without loss of generality, we can compute $\mathbf{U}_{\rho(x)} := \mathbf{U}_{\rho(x),1} \mathbf{B}_1^{*T} + \mathbf{U}_{\rho(x),2} \mathbf{B}_2^{*T} + \mathbf{U}_{\rho(x),3} \mathbf{B}_3^{*T}$, where $\mathbf{U}_{\rho(x),1} \xleftarrow{\$} \mathbb{Z}_q^{(k+1) \times k}$, $\mathbf{U}_{\rho(x),2} \xleftarrow{\$} \mathbb{Z}_q^{(k+1) \times k}$, and $\mathbf{U}_{\rho(x),3} \xleftarrow{\$} \mathbb{Z}_q^{(k+1) \times 1}$. Let the first entry of $\mathbf{M}_x$ be m_x, that is, $\mathbf{M}_x = (m_x, \ldots)$. Then, observe that we can write

$$
\begin{aligned}
\mathbf{M}_x \begin{pmatrix} \gamma \mathbf{B}_3^{*T} \\ \mathbf{W}_{\mathsf{bot}} \end{pmatrix} + \mathbf{c}_x^T \mathbf{U}_{\rho(x)} &= \mathbf{M}_x \begin{pmatrix} \mathbf{0} \\ \mathbf{W}_{\mathsf{bot}} \end{pmatrix} + \mathbf{M}_x \begin{pmatrix} \gamma \mathbf{B}_3^{*T} \\ \mathbf{0} \end{pmatrix} + \mathbf{c}_x^T \mathbf{U}_{\rho(x)} \\
&= \mathbf{M}_x \begin{pmatrix} \mathbf{0} \\ \mathbf{W}_{\mathsf{bot}} \end{pmatrix} + (m_x, \ldots) \begin{pmatrix} \gamma \mathbf{B}_3^{*T} \\ \mathbf{0} \end{pmatrix} + \mathbf{c}_x^T \mathbf{U}_{\rho(x)} \\
&= \mathbf{M}_x \begin{pmatrix} \mathbf{0} \\ \mathbf{W}_{\mathsf{bot}} \end{pmatrix} + m_x \gamma \mathbf{B}_3^{*T} + \mathbf{c}_x^T \mathbf{U}_{\rho(x)} \\
&= \mathbf{M}_x \begin{pmatrix} \mathbf{0} \\ \mathbf{W}_{\mathsf{bot}} \end{pmatrix} + (m_x \gamma + \mathbf{c}_x^T \mathbf{U}_{\rho(x),3}) \mathbf{B}_3^{*T} + \mathbf{c}_x^T \mathbf{U}_{\rho(x),1} \mathbf{B}_1^{*T} + \mathbf{c}_x^T \mathbf{U}_{\rho(x),2} \mathbf{B}_2^{*T} \\
&\equiv \mathbf{M}_x \begin{pmatrix} \mathbf{0} \\ \mathbf{W}_{\mathsf{bot}} \end{pmatrix} + (\mathbf{c}_x^T \mathbf{U}'_{\rho(x),3}) \mathbf{B}_3^{*T} + \mathbf{c}_x^T \mathbf{U}_{\rho(x),1} \mathbf{B}_1^{*T} + \mathbf{c}_x^T \mathbf{U}_{\rho(x),2} \mathbf{B}_2^{*T}
\end{aligned}
$$

where we can write $\mathbf{U}'_{\rho(x),3} = \mathbf{U}_{\rho(x),3} + \Delta$ such that $m_x \gamma = \mathbf{c}_x^T \Delta$. Therefore, to complete the proof, it suffices to argue that $\mathbf{U}_{\rho(x),3}$ and $\mathbf{U}'_{\rho(x),3}$ are identically distributed. We show this next. Observe that since ρ is injective, hence it follows that $\mathbf{U}_{\rho(x)}$ is a fresh random matrix and the only other place it appears is in secret keys $\mathsf{sk}_{\rho(x),\mathsf{GID}}$. We argue that $\mathsf{sk}_{\rho(x),\mathsf{GID}}$ information theoretically leaks no information about $\mathbf{U}_{\rho(x),3}$ and hence $\mathbf{U}_{\rho(x),3}$ and $\mathbf{U}'_{\rho(x),3}$ are identically distributed. To see this, observe that $\mathsf{sk}_{\rho(x),\mathsf{GID}}$ information theoretically reveals $\mathbf{V}_{\rho(x)} \mathbf{k} + \mathbf{U}_{\rho(x)} \mathbf{B}_1 \mathbf{h}_{\mathsf{GID}}$, where $\mathbf{U}_{\rho(x)} \mathbf{B}_1 \mathbf{h}_{\mathsf{GID}} = \mathbf{U}_{\rho(x),1} \mathbf{h}_{\mathsf{GID}}$ since $\mathbf{B}_1^{*T} \mathbf{B}_1 = \mathbf{I}$, $\mathbf{B}_2^{*T} \mathbf{B}_1 = \mathbf{0}$ and $\mathbf{B}_3^{*T} \mathbf{B}_1 = \mathbf{0}$, thus no information about $\mathbf{U}_{\rho(x),3}$ is revealed.

To complete the proof, we argue that substituting $\mathbf{U}_{\rho(x),3}$ with $\mathbf{U}'_{\rho(x),3}$ (as described above) for all rows x of matrix $\mathbf{M}$ for which the challenger sampled the authority keys (that is, uncorrupted

rows plus the rows for which the adversary queried the master secret key) allows us to move from Hyb_2 to Hyb_1. We have already argued that this substitution does not change the distribution of the secret keys and ciphertext obtained by the adversary $\mathcal{A}$ for the uncorrupted rows of $\mathbf{M}$. For the case of rows x of $\mathbf{M}$ for which the adversary queried the master secret key, the adversary additionally learns $\mathbf{U}'_{\rho(x),3}$ and $\mathbf{M}_x \begin{pmatrix} \mathbf{0}^T \\ \mathbf{W}_{\mathsf{bot}} \end{pmatrix}$ in Hyb_1 and $\mathbf{U}_{\rho(x),3}$ and $\mathbf{M}_x \begin{pmatrix} \gamma \mathbf{B}_3^{*T} \\ \mathbf{W}_{\mathsf{bot}} \end{pmatrix}$ in Hyb_2 and we argue that this does not help the adversary $\mathcal{A}$ to distinguish between Hyb_1 and Hyb_2. This is because $\mathbf{U}_{\rho(x),3}$ and $\mathbf{U}'_{\rho(x),3}$ are identically distributed and $\mathbf{M}_x \begin{pmatrix} \gamma \mathbf{B}_3^{*T} \\ \mathbf{W}_{\mathsf{bot}} \end{pmatrix} = \mathbf{M}_x \begin{pmatrix} \mathbf{0}^T \\ \mathbf{W}_{\mathsf{bot}} \end{pmatrix}$ due to the game condition.

Therefore, it follows that Hyb_1 and Hyb_2 are statistically indistinguishable. This completes the proof of Claim 4.5. □

Proof of Claim 4.6.

Suppose towards a contradiction that there exists a p.p.t. adversary $\mathcal{A}$ that distinguishes its view in $\mathsf{Hyb}_{3,j-1}$ and $\mathsf{Hyb}_{3,j,1}$ with a noticeable advantage ϵ. Then we show how to construct an adversary $\mathcal{B}$ using $\mathcal{A}$ that breaks the $\mathsf{SD}^{\mathbb{G}_2}_{\mathbf{B}_1 \to \mathbf{B}_1, \mathbf{B}_2}$ assumption with noticeable advantage ϵ. Suppose that the $\mathsf{SD}^{\mathbb{G}_2}_{\mathbf{B}_1 \to \mathbf{B}_1, \mathbf{B}_2}$ challenger is $\mathcal{C}$. Then, recall that $\mathcal{B}$ obtains $(\mathcal{D}, [\![\mathbf{h}]\!]_2)$ from $\mathcal{C}$ and $\mathcal{B}$'s goal is to distinguish whether $\mathbf{h} \in \mathsf{span}(\mathbf{B}_1)$ or $\mathbf{h} \in \mathsf{span}(\mathbf{B}_1, \mathbf{B}_2)$, where

$$
\begin{aligned}
&\mathcal{PG} := (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, q, g_1, g_2, g_T) \leftarrow \mathsf{PGGen}(1^\lambda), \\
&\mathbf{B}_1 \xleftarrow{\$} \mathbb{Z}_q^{(2k+1)\times k}, \mathbf{B}_2 \xleftarrow{\$} \mathbb{Z}_q^{(2k+1)\times k}, \mathbf{B}_3 \xleftarrow{\$} \mathbb{Z}_q^{(2k+1)\times 1}, \\
&(\mathbf{B}_1^*, \mathbf{B}_2^*, \mathbf{B}_3^*)^T = (\mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3)^{-1}, \\
&\mathcal{D} = (\mathcal{PG}, [\![\mathbf{B}_1]\!]_2, [\![\mathbf{B}_2]\!]_2, [\![\mathbf{B}_3]\!]_2, \mathsf{basis}(\mathbf{B}_1^*), \mathsf{basis}(\mathbf{B}_3^*), \mathsf{basis}(\mathbf{B}_1^*, \mathbf{B}_2^*)).
\end{aligned}
$$

$\mathcal{B}$ accomplishes this by invoking the adversary $\mathcal{A}$ for distinguishing $\mathsf{Hyb}_{3,j-1}$ and $\mathsf{Hyb}_{3,j,1}$. Towards this, $\mathcal{B}$ plays the role of $\mathcal{A}$'s challenger in its dinstinguishing game as follows: instead of running $\mathsf{GlobalSetup}^*$ on its own completely, it uses $[\![\mathbf{B}_1]\!]_2, [\![\mathbf{B}_2]\!]_2, [\![\mathbf{B}_3]\!]_2, \mathsf{basis}(\mathbf{B}_1^*), \mathsf{basis}(\mathbf{B}_3^*), \mathsf{basis}(\mathbf{B}_1^*, \mathbf{B}_2^*)$ obtained from $\mathcal{C}$ and samples $\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_1^*, \mathbf{A}_2^*$ on its own as in $\mathsf{GlobalSetup}^*$. Consequently, gp is identically distributed as in $\mathsf{GlobalSetup}^*$. While $\mathcal{B}$ does not know $\mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3, \mathbf{B}_1^*, \mathbf{B}_2^*, \mathbf{B}_3^*$ completely now, we show that the information obtained from $\mathcal{C}$ in this regard is sufficient for $\mathcal{B}$ to perfectly simulate the view for $\mathcal{A}$. This information shows up in two places in the game between $\mathcal{B}$ and $\mathcal{A}$:

- When computing challenge ciphertext, $\mathcal{B}$ needs to compute matrix $\mathbf{W}$ whose first row is $\gamma \mathbf{B}_3^{*T}$. Observe that since $\mathbf{B}_3^*$ is a vector, hence $\mathsf{basis}(\mathbf{B}_3^*) = \delta \mathbf{B}_3^*$ for some non-zero $\delta \in \mathbb{Z}_q$. Therefore, $\mathcal{B}$ can sample $\gamma' \xleftarrow{\$} \mathbb{Z}_q$ and set the first row of $\mathbf{W}$ to be $\gamma' \mathsf{basis}(\mathbf{B}_3^*)^T$. This perfectly simulates $\gamma \mathbf{B}_3^{*T}$ as it implicitly sets $\gamma = \gamma' \delta$ and γ is uniform random since γ' is uniform random.
- For computing H_1 on some input $\mathsf{GID}_{\mathsf{index}}$, $\mathcal{B}$ can use $[\![\mathbf{B}_1]\!]_2$, $[\![\mathbf{B}_3]\!]_2$ and $[\![\mathbf{h}]\!]_2$ obtained from $\mathcal{C}$ as follows:

$$
\mathbf{h}_{\mathsf{GID}_{\mathsf{index}}} \xleftarrow{\$} \mathbb{Z}_q^k, \mathsf{H}_1(\mathsf{GID}_{\mathsf{index}}) := \begin{cases} ([\![\mathbf{B}_1]\!]_2 \odot \mathbf{h}_{\mathsf{GID}_{\mathsf{index}}}) \cdot [\![\mathbf{B}_3]\!]_2 & , \text{if } \mathsf{index} \leq j-1 \\ [\![\mathbf{h}]\!]_2 & , \text{if } \mathsf{index} = j \\ [\![\mathbf{B}_1]\!]_2 \odot \mathbf{h}_{\mathsf{GID}_{\mathsf{index}}} & , \text{if } \mathsf{index} > j \end{cases}.
$$

Therefore, it follows that when $\mathbf{h} \in \mathsf{span}(\mathbf{B}_1)$, $\mathcal{B}$ perfectly simulates $\mathsf{Hyb}_{3,j-1}$ to $\mathcal{A}$ and when $\mathbf{h} \in \mathsf{span}(\mathbf{B}_1, \mathbf{B}_2)$, $\mathcal{B}$ perfectly simulates $\mathsf{Hyb}_{3,j,1}$ to $\mathcal{A}$. Therefore, $\mathcal{B}$'s winning advantage against $\mathcal{C}$ is same as $\mathcal{A}$'s winning advantage ϵ against $\mathcal{B}$. But this contradicts the assumption that such a p.p.t. adversary $\mathcal{B}$ cannot exist. This completes the proof of Claim 4.6. □

Proof of Claim 4.7. Observe that the only difference between $\mathsf{Hyb}_{3,j,1}$ and $\mathsf{Hyb}'_{3,j,1}$ is that in ciphertext component $c_{3,x}$ for all $x \in [n]$: $c_{3,x}$ contains $[\![\omega_x]\!]_1$, where ω_x is a secret share of $\gamma \mathbf{B}_3^{*T} \in \mathbb{Z}_q^{1\times(2k+1)}$ in $\mathsf{Hyb}_{3,j,1}$, but it is a secret share of $(\mathbf{B}_2^*\boldsymbol{\delta} + \gamma\mathbf{B}_3^*)^T$ in $\mathsf{Hyb}'_{3,j,1}$. Therefore, to prove that the hybrids are statistically indistinguishable, we will argue that $\mathbf{B}_2^*\boldsymbol{\delta}$ is information theoretically hidden to the adversary $\mathcal{A}$ in $\mathsf{Hyb}'_{3,j,1}$.

Suppose the challenge access policy $(\mathbf{M}, \rho)$ is defined over a set of authorities $U \subseteq \mathcal{AU}$, that is, $\rho : [n] \to U$. Recall from Appendix A.5 that the game condition requires that $U_\mathcal{A} \cap U_\mathcal{B} = \emptyset$ and for each $\mathsf{GID} \in \mathcal{GID}$, it is required that $S \cup S_{\mathsf{GID}} \notin (\mathbf{M}, \rho)$, where

- $U_\mathcal{A} \subseteq U$ denotes the set of attribute authorities for which the adversary supplied the authority public keys $\{\mathsf{pk}_i\}$,
- $U_\mathcal{B}$ denote the set of attribute authorities for which the challenger created the master public key pairs in response to the authority setup query of the adversary so far,
- S denotes the subset of U containing the authorities in $U_\mathcal{A}$ plus the authorities for which the adversary made a master key query so far (in other words, S denotes the set of corrupt authorities),
- for each global identifier $\mathsf{GID} \in \mathcal{GID}$, S_{GID} denotes the subset of U containing authorities i such that the adversary queried a secret key for the pair (i, GID).

To show that $\mathbf{B}_2^*\boldsymbol{\delta}$ is information theoretically hidden from the adversary $\mathcal{A}$ in $\mathsf{Hyb}'_{3,j,1}$, we only need to rely on the second game condition and that too only for the j^{th} GID, that is, we will use the fact that $S \cup S_{\mathsf{GID}_j} \notin (\mathbf{M}, \rho)$. Here, $S \cup S_{\mathsf{GID}_j} \notin (\mathbf{M}, \rho)$ is a shorthand for $(1, 0, \ldots, 0) \notin \mathsf{rowSpan}(\{\mathbf{M}_x\}_{\rho(x)\in S\cup S_{\mathsf{GID}_j}})$.

Note that the vectors $\mathbf{M}_x\mathbf{W}$ for all rows x of the challenge access matrix $\mathbf{M}$ labeled by corrupt authorities (that is, $\rho(x) \in S$) are information theoretically revealed to $\mathcal{A}$. However, by the game condition the subspace spanned by those rows does not include the vector $(1, 0, \ldots, 0)$. This means that there must exist some vector $\mathbf{u} \in \mathbb{Z}_q^\ell$ such that $\mathbf{u}$ is orthogonal to all these rows of $\mathbf{M}$ (that is $\mathbf{M}_x\mathbf{u} = 0$) but is not orthogonal to $(1, 0, \ldots, 0)$, that is, the first entry of $\mathbf{u}$ must be non-zero. We consider a basis $\mathbb{U}$ of $\mathbb{Z}_q^\ell$ involving the vector $\mathbf{u}$ and write $\mathbf{W} = \begin{pmatrix} (\mathbf{B}_2^*\boldsymbol{\delta} + \gamma\mathbf{B}_3^*)^T \\ \mathbf{W}_{\mathsf{bot}} \end{pmatrix} = \widetilde{\mathbf{U}} + \mathbf{u}\mathbf{b}^T$ for some $\mathbf{b} \in \mathbb{Z}_q^{2k+1}$ and some $\widetilde{\mathbf{U}} \in \mathbb{Z}_q^{\ell\times(2k+1)}$ such that each column of $\widetilde{\mathbf{U}}$ lies in the column span of $\mathbb{U} \setminus \mathbf{u}$. Hence, $\widetilde{\mathbf{U}}$ reveals no information about $\mathbf{b}$. Now since the first entry of $\mathbf{u}$ is non-zero, it follows that the first row of $\mathbf{W}$, that is, $(\mathbf{B}_2^*\boldsymbol{\delta} + \gamma\mathbf{B}_3^*)^T$, depends on $\mathbf{b}$. But $\mathbf{M}_x\mathbf{W}$ for all the corrupted rows of $\mathbf{M}$ contains no information about $\mathbf{b}$ since $\mathbf{u}$ is orthogonal to all these rows. Thus, it follows that these rows do not leak information about $(\mathbf{B}_2^*\boldsymbol{\delta} + \gamma\mathbf{B}_3^*)^T$.

Therefore, the only possible way for $\mathcal{A}$ to get information about $\mathbf{B}_2^*\boldsymbol{\delta}$ is through the ciphertext components $c_{3,x}$ corresponding to the uncorrupted rows of $\mathbf{M}$. However, for each such row x, $\mathcal{A}$ can only recover $\mathbf{c}_x^T$, $\mathbf{M}_x\mathbf{W} + \mathbf{c}_x^T\mathbf{U}_{\rho(x)}$ information theoretically. For the uncorrupted rows x, we analyze two cases:

- x **such that** $\rho(x) \in S_{\mathsf{GID}_j}$**:** These are uncorrupted rows of $\mathbf{M}$ labeled by authorities who issued key for GID_j. From the game condition, it follows that $(1, 0, \ldots, 0) \notin \mathsf{rowSpan}(\{\mathbf{M}_x\}_{\rho(x)\in S_{\mathsf{GID}_j}})$. Therefore, using the same argument as for the analysis for rows corresponding to corrupt authorities above, it follows that $\mathbf{M}_x\mathbf{W}$ contains no information about $(\mathbf{B}_2^*\boldsymbol{\delta} + \gamma\mathbf{B}_3^*)^T$. Hence, $\mathbf{c}_x^T$, $\mathbf{M}_x\mathbf{W} + \mathbf{c}_x^T\mathbf{U}_{\rho(x)}$ reveals no information about $(\mathbf{B}_2^*\boldsymbol{\delta} + \gamma\mathbf{B}_3^*)^T$ to the adversary $\mathcal{A}$.

- x **such that** $\rho(x) \in U \setminus (S \cup S_{\mathsf{GID}_j})$**:** These are uncorrupted rows of $\mathbf{M}$ labeled by authorities who are neither corrupt nor issued key for GID_j. The game condition does not apply for these rows and the analysis requires more care as described next. Without loss of generality, we can compute $\mathbf{U}_{\rho(x)} := \mathbf{U}_{\rho(x),1}\mathbf{B}_1^{*T} + \mathbf{U}_{\rho(x),2}\mathbf{B}_2^{*T} + \mathbf{U}_{\rho(x),3}\mathbf{B}_3^{*T}$, where $\mathbf{U}_{\rho(x),1} \xleftarrow{\$} \mathbb{Z}_q^{(k+1)\times k}$, $\mathbf{U}_{\rho(x),2} \xleftarrow{\$} \mathbb{Z}_q^{(k+1)\times k}$, and $\mathbf{U}_{\rho(x),3} \xleftarrow{\$} \mathbb{Z}_q^{(k+1)\times 1}$. Let the first entry of $\mathbf{M}_x$ be m_x, that is, $\mathbf{M}_x = (m_x, \ldots)$. Then, observe that we can write

$$
\begin{aligned}
&\mathbf{M}_x \begin{pmatrix} (\mathbf{B}_2^*\boldsymbol{\delta} + \gamma\mathbf{B}_3^*)^T \\ \mathbf{W}_{\mathsf{bot}} \end{pmatrix} + \mathbf{c}_x^T\mathbf{U}_{\rho(x)} \\
&= \mathbf{M}_x \begin{pmatrix} \gamma\mathbf{B}_3^{*T} \\ \mathbf{W}_{\mathsf{bot}} \end{pmatrix} + (m_x, \ldots) \begin{pmatrix} (\mathbf{B}_2^*\boldsymbol{\delta})^T \\ \mathbf{0} \end{pmatrix} + \mathbf{c}_x^T\mathbf{U}_{\rho(x)} \\
&= \mathbf{M}_x \begin{pmatrix} \gamma\mathbf{B}_3^{*T} \\ \mathbf{W}_{\mathsf{bot}} \end{pmatrix} + m_x(\mathbf{B}_2^*\boldsymbol{\delta})^T + \mathbf{c}_x^T\mathbf{U}_{\rho(x)} \\
&= \mathbf{M}_x \begin{pmatrix} \gamma\mathbf{B}_3^{*T} \\ \mathbf{W}_{\mathsf{bot}} \end{pmatrix} + (m_x\boldsymbol{\delta}^T + \mathbf{c}_x^T\mathbf{U}_{\rho(x),2})\mathbf{B}_2^{*T} + \mathbf{c}_x^T(\mathbf{U}_{\rho(x),1}\mathbf{B}_1^{*T} + \mathbf{U}_{\rho(x),3}\mathbf{B}_3^{*T}) \\
&= \mathbf{M}_x \begin{pmatrix} \gamma\mathbf{B}_3^{*T} \\ \mathbf{W}_{\mathsf{bot}} \end{pmatrix} + (\mathbf{c}_x^T\mathbf{U}'_{\rho(x),2})\mathbf{B}_2^{*T} + \mathbf{c}_x^T(\mathbf{U}_{\rho(x),1}\mathbf{B}_1^{*T} + \mathbf{U}_{\rho(x),3}\mathbf{B}_3^{*T})
\end{aligned}
$$

  where we can write $\mathbf{U}'_{\rho(x),2} = \mathbf{U}_{\rho(x),2} + \Delta$ such that $m_x\boldsymbol{\delta}^T = \mathbf{c}_x^T\Delta$. Therefore, to complete the proof, it suffices to argue that $\mathbf{U}_{\rho(x),2}$ and $\mathbf{U}'_{\rho(x),2}$ are identically distributed. We show this next.

  Observe that since ρ is injective, hence it follows that $\mathbf{U}_{\rho(x)}$ is a fresh random matrix and the only other place it appears is in secret keys $\mathsf{sk}_{\rho(x),\mathsf{GID}_{\mathsf{index}}}$. We argue that $\mathsf{sk}_{\rho(x),\mathsf{GID}_{\mathsf{index}}}$ information theoretically reveals no information about $\mathbf{U}_{\rho(x),2}$ to the adversary $\mathcal{A}$ and hence $\mathbf{U}_{\rho(x),2}$ and $\mathbf{U}'_{\rho(x),2}$ are identically distributed. To see this, observe that the $\mathbf{U}_{\rho(x)}$-dependent term of $\mathsf{sk}_{\rho(x),\mathsf{GID}_{\mathsf{index}}}$ is of the form $\mathbf{U}_{\rho(x)} \odot \mathsf{H}_1(\mathsf{GID}_{\mathsf{index}})$, where $\mathsf{H}_1(\mathsf{GID}_{\mathsf{index}})$ is of the following form depending on index: $[\![\mathbf{B}_1\mathbf{h}_{\mathsf{GID}_{\mathsf{index}}} + \mathbf{B}_3]\!]_2$ if $\mathsf{index} \le j-1$, $[\![\mathbf{B}_1\mathbf{h}_{\mathsf{GID}_{\mathsf{index}}} + \mathbf{B}_2\mathbf{h}'_{\mathsf{GID}_{\mathsf{index}}}]\!]_2$ if $\mathsf{index} = j$, and $[\![\mathbf{B}_1\mathbf{h}_{\mathsf{GID}_{\mathsf{index}}}]\!]_2$ if $\mathsf{index} > j$. We analyze the three cases separately:

  - **Case 1:** $\mathsf{index} < j$**.** Observe that the $\mathbf{U}_{\rho(x)}$-dependent term of $\mathsf{sk}_{\rho(x),\mathsf{GID}_{\mathsf{index}}}$ information theoretically reveals $\mathbf{U}_{\rho(x),1}\mathbf{h}_{\mathsf{GID}_{\mathsf{index}}} + \mathbf{U}_{\rho(x),3}$ since $\mathbf{B}_2^{*T}\mathbf{B}_1 = \mathbf{0}$ and $\mathbf{B}_2^{*T}\mathbf{B}_3 = \mathbf{0}$.
  - **Case 2:** $\mathsf{index} = j$**.** This case requires no analysis since adversary $\mathcal{A}$ never sees secret keys $\mathsf{sk}_{\rho(x),\mathsf{GID}_j}$. This is due to the definition of set S_{GID_j} and the fact that we are only considering x such that $\rho(x) \in U \setminus (S \cup S_{\mathsf{GID}_j})$.
  - **Case 3:** $\mathsf{index} > j$**.** Observe that the $\mathbf{U}_{\rho(x)}$-dependent term of $\mathsf{sk}_{\rho(x),\mathsf{GID}_{\mathsf{index}}}$ information theoretically reveals $\mathbf{U}_{\rho(x),1}\mathbf{h}_{\mathsf{GID}_{\mathsf{index}}}$ since $\mathbf{B}_2^{*T}\mathbf{B}_1 = \mathbf{0}$.

  Hence, it follows that $\mathsf{sk}_{\rho(x),\mathsf{GID}_{\mathsf{index}}}$ information theoretically reveals no information about $\mathbf{U}_{\rho(x),2}$, so $\mathbf{U}_{\rho(x)}$ and $\mathbf{U}'_{\rho(x)}$ are identically distributed. Thus, for all x such that $\rho(x) \in U \setminus (S \cup S_{\mathsf{GID}_j})$, ciphertext components $c_{3,x}$ reveal no information about $\mathbf{B}_2^*\boldsymbol{\delta}$ to the adversary $\mathcal{A}$.

To complete the proof, we argue that substituting $\mathbf{U}_{\rho(x),2}$ with $\mathbf{U}'_{\rho(x),2}$ (as described above) for all rows x of matrix $\mathbf{M}$ for which the challenger sampled the authority keys (that is, uncorrupted rows plus the rows for which the adversary queried the master secret key) allows us to move from $\mathsf{Hyb}'_{3,j,1}$ to $\mathsf{Hyb}_{3,j,1}$. We have already argued that this substitution does not change the distribution of the

secret keys and ciphertext obtained by the adversary $\mathcal{A}$ for the uncorrupted rows of $\mathbf{M}$. For the case of rows x of $\mathbf{M}$ for which the adversary queried the master secret key, the adversary additionally learns $\mathbf{U}'_{\rho(x),2}$ and $\mathbf{M}_x \begin{pmatrix} \gamma \mathbf{B}_3^{*T} \\ \mathbf{W}_{\mathsf{bot}} \end{pmatrix}$ in $\mathsf{Hyb}_{3,j,1}$ and $\mathbf{U}_{\rho(x),2}$ and $\mathbf{M}_x \begin{pmatrix} (\mathbf{B}_2^* \boldsymbol{\delta} + \gamma \mathbf{B}_3^*)^T \\ \mathbf{W}_{\mathsf{bot}} \end{pmatrix}$ in $\mathsf{Hyb}'_{3,j,1}$ and we argue that this does not help the adversary $\mathcal{A}$ to distinguish between $\mathsf{Hyb}_{3,j,1}$ and $\mathsf{Hyb}'_{3,j,1}$. This is because $\mathbf{U}_{\rho(x),2}$ and $\mathbf{U}'_{\rho(x),2}$ are identically distributed and $\mathbf{M}_x \begin{pmatrix} \gamma \mathbf{B}_3^{*T} \\ \mathbf{W}_{\mathsf{bot}} \end{pmatrix} = \mathbf{M}_x \begin{pmatrix} (\mathbf{B}_2^* \boldsymbol{\delta} + \gamma \mathbf{B}_3^*)^T \\ \mathbf{W}_{\mathsf{bot}} \end{pmatrix}$ due to the game condition.

From the case analysis above, it follows that $\mathsf{Hyb}_{3,j,1}$ and $\mathsf{Hyb}'_{3,j,1}$ are statistically indistinguishable. This completes the proof of Claim 4.7.

□

Proof of Claim 4.8. Similar to proof of Claim 4.6. □

Proof of Claim 4.9. Similar to the proof of Claim 4.5. □

Proof of Claim 4.10. Similar to proof of Claim 4.6. □

Proof of Claim 4.13. This holds since Hyb_5 contains no information about msg_0 and msg_1. □

C MA-ABE for NC^1 with Multi-Use Security: Appendix

C.1 Core 1-ABE

Definition C.1 (Core 1-ABE Games $\mathsf{G}_{\mathcal{A}}^{\mathsf{1\text{-}ABE},0}$, $\mathsf{G}_{\mathcal{A}}^{\mathsf{1\text{-}ABE},1}$ [KW19])**.** *For a stateful adversary $\mathcal{A}$, for $b \in \{0,1\}$, define the game $\mathsf{G}_{\mathcal{A}}^{\mathsf{1\text{-}ABE},b}$ as*

$$\begin{aligned} &\mathbf{u}_i \leftarrow \mathsf{CPA.Setup}(1^\lambda), \\ &(\mu^{(0)}, \mu^{(1)}) \xleftarrow{\$} \mathbb{Z}_q, \\ &b' \leftarrow \mathcal{A}^{\mathcal{O}_X(\cdot), \mathcal{O}_E(\cdot,\cdot), \mathcal{O}_F(\cdot)}(\mu^{(0)}), \\ &\textbf{ret } b' \end{aligned}$$

where $\mathcal{O}_F(f) = \mathsf{ct} := \{\mathsf{sk}'_f = \{\mu_j\}_{\rho'(j)=0} \cup \{\mathsf{CPA.Enc}(\mathbf{u}_{\rho'(j)}, \mu_j)\}_{\rho'(j)\neq 0}\}$, where $(\{\mu_j\}, \rho') \leftarrow \mathsf{LSSS.Share}(f, \mu^{(b)})$, and $\mathcal{O}_X(x) := (\mathsf{ct}'_x = \{\mathbf{u}_i\}_{x_i=1})$, and $\mathcal{O}_E(i,m) := \mathsf{CPA.Enc}(\mathbf{u}_i, m)$, with the restriction that (i) only one query is made to each $\mathcal{O}_F(\cdot)$ and $\mathcal{O}_X(\cdot)$, and (ii) the queries f and x to $\mathcal{O}_F(\cdot,\cdot)$ and $\mathcal{O}_X(\cdot)$ respectively satisfy $f(x) = 0$.

To be clear, the b in the game $\mathbb{G}_{\mathcal{A}}^{\mathsf{1\text{-}ABE},b}$ affects only the implementation of the $\mathcal{O}_F(\cdot)$ oracle (where $\mu^{(b)}$ is shared), and $\mathcal{A}$ is given $\mu^{(0)}$ as input in *both* games.

The CPA-secure symmetric encryption scheme in [KW19] is constructed as follows:

$\mathsf{CPA.Setup}(1^\lambda)$:

1 : Run $\mathbb{G} \leftarrow \mathcal{G}(1^\lambda)$. Sample $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^k$. Output $\mathsf{sk} := \mathbf{u}$.

$\mathsf{CPA.Enc}(\mathsf{sk}, [\![m]\!])$:

1 : Sample $\mathbf{r} \xleftarrow{\$} \mathbb{Z}_q^k$. Output $\mathsf{ct} := (\mathsf{ct}_0 = [\![\mathbf{r}]\!], \mathsf{ct}_1 = [\![m + \mathbf{u}^T\mathbf{r}]\!])$.

$\mathsf{CPA.Dec}(\mathsf{sk}, \mathsf{ct})$:

1 : Output $\mathsf{ct}_1 / (\mathsf{sk}^T \odot \mathsf{ct}_0)$.

Figure 4: CPA-secure symmetric encryption scheme in [KW19]

Lemma C.2 (Core 1-ABE security [KW19]). *For the* Core 1-ABE *component of Definition C.1 implemented with the CPA-secure symmetric encryption scheme* (CPA.Setup, CPA.Enc, CPA.Dec) *from Figure 4, the following is a negligible function in λ when the policy class is* NC^1*:*

$$|\Pr[\mathsf{G}_{\mathcal{A}}^{\text{1-ABE},0}(\lambda)=1]-\Pr[\mathsf{G}_{\mathcal{A}}^{\text{1-ABE},1}(\lambda)=1]|$$

C.2 Proof of Theorem 5.1

We prove full adaptive security along with multi-use support (that is, ρ is not required to be injective) of the MA-ABE scheme for NC^1 presented in Figure 1 along with the amended dimension changes specified in Section 5.1. We first formally state Theorem 5.1 next.

Theorem C.3. *The MA-ABE construction in Figure 1 amended with dimension changes specified in Section 5.1 supports* NC^1 *circuits and is fully adaptively secure (Definition A.6) if all of the following hold true.*

- *game condition holds.*
- *k-MDDH assumption holds in groups $\mathbb{G}_1$ and $\mathbb{G}_2$ (Assumption A.8).*
- $\mathsf{SD}^{\mathbb{G}_2}_{\mathbf{B}_1\to\mathbf{B}_1,\mathbf{B}_2}$ *assumption holds (Assumption A.9).*
- $\mathsf{SD}^{\mathbb{G}_2}_{\mathbf{B}_2\to\mathbf{B}_2,\mathbf{B}_3}$ *assumption holds (Assumption A.9).*

We prove this via a sequence of hybrid games. Suppose the adversary makes q number of queries to the random oracle H_1. Then, the hybrid games are as follows:

$$\mathsf{Hyb}_{\mathsf{Real}}, \mathsf{Hyb}'_{\mathsf{Real}}, \mathsf{Hyb}_1, \mathsf{Hyb}_2, \{\mathsf{Hyb}_{3,j,1}, \mathsf{Hyb}'_{3,j,1}, \mathsf{Hyb}_{3,j,2}, \mathsf{Hyb}'_{3,j,2}, \mathsf{Hyb}_{3,j,3}\}_{j\in[q]}, \mathsf{Hyb}_4, \mathsf{Hyb}_5.$$

The hybrid descriptions are exactly the same as in Section 4.1. The main difference is that each of the statistical indistinguishability steps are replaced by computational indistinguishability, where instead of relying on the injectivity of ρ, we will reduce the indistinguishability to the security ofthe Core 1-ABE construction of [KW19]. In more detail, we will use the security of the Core 1-ABE to prove the indistinguishability of the following pairs of distributions:

- Hyb_1 and Hyb_2,
- $\mathsf{Hyb}_{3,j,1}$ and $\mathsf{Hyb}'_{3,j,1}$ for $j\in[q]$,
- $\mathsf{Hyb}_{3,j,2}$ and $\mathsf{Hyb}'_{3,j,2}$ for $j\in[q]$,
- $\mathsf{Hyb}_{3,q}$ and Hyb_4.

Note that the security of Core 1-ABE is not an additional assumption required since we already assumed k-MDDH.

Before proceeding to prove the above transitions, we note that the change of parameter dimensions in the construction slightly affects the parameter dimensions in $\mathsf{GlobalSetup}^*$ algorithm used from hybrid $\mathsf{Hyb}'_{\mathsf{Real}}$ onwards as follows. We describe $\mathsf{GlobalSetup}^*$ next and highlight the changes of dimensions.

$\mathsf{GlobalSetup}^*$ runs the same computation as $\mathsf{GlobalSetup}$ to compute gp and additionally also computes the following:

$$\mathbf{A}_2 \xleftarrow{\$} \mathbb{Z}_q^{\boxed{k} \times \boxed{2k}}, \mathbf{B}_1, \mathbf{B}_2 \xleftarrow{\$} \mathbb{Z}_q^{(2k+1)\times k}, \mathbf{B}_3 \xleftarrow{\$} \mathbb{Z}_q^{(2k+1)\times 1}$$
$$\left(\mathbf{A}_1^*, \mathbf{A}_2^*\right) = \begin{pmatrix}\mathbf{A}_1\\ \mathbf{A}_2\end{pmatrix}^{-1}, \left(\mathbf{B}_1^*, \mathbf{B}_2^*, \mathbf{B}_3^*\right) = \left(\left(\mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3\right)^{-1}\right)^T.$$

Let $\mathbf{A} = \begin{pmatrix}\mathbf{A}_1\\ \mathbf{A}_2\end{pmatrix}$, $\mathbf{A}^* = \left(\mathbf{A}_1^*, \mathbf{A}_2^*\right)$, $\mathbf{B} = \left(\mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3\right)$, $\mathbf{B}^* = \left(\mathbf{B}_1^*, \mathbf{B}_2^*, \mathbf{B}_3^*\right)$. Then, $\mathsf{st} = (\mathbf{A}, \mathbf{A}^*, \mathbf{B}, \mathbf{B}^*)$.

Previously, the dimensions of $\mathbf{A}_1^*$ were $(k+1) \times k$, now they are $\boxed{2k} \times k$. Previously, the dimensions of $\mathbf{A}_2^*$ were $(k+1) \times 1$, now they are $\boxed{2k} \times \boxed{k}$. Observe that $\forall i, j \in \{1,2\}$: $\mathbf{A}_i \mathbf{A}_j^* = \mathbf{I}$ if $i = j$, and $\mathbf{0}$ if $i \neq j$. Similarly, $\forall i, j \in \{1,2,3\}$: $\mathbf{B}_i^T \mathbf{B}_j^* = \mathbf{I}$ if $i = j$, and $\mathbf{0}$ if $i \neq j$.

Next we prove the aforementioned transitioned through a series of claims.

Claim C.4. *If the game condition holds and* $\mathsf{Core\ 1\text{-}ABE}$ *is secure, then,* Hyb_1 *and* Hyb_2 *are computationally indistinguishable.*

Proof. Observe that the only difference between Hyb_1 and Hyb_2 is that in ciphertext component $c_{3,x}$ for all $x \in [n]$: $c_{3,x}$ contains $[\![\omega_x]\!]_1$, where ω_x is a secret share of $\mathbf{0}^T \in \mathbb{Z}_q^{1\times(2k+1)}$ in Hyb_1, but it is a secret share of $\gamma \mathbf{B}_3^{*T}$ in Hyb_2. Therefore, to prove that the hybrids are computationally indistinguishable, we will reduce the indistinguishability to the security of $\mathsf{Core\ 1\text{-}ABE}$.

Suppose that there exists an adversary $\mathcal{A}$ that can distinguish between Hyb_1 and Hyb_2 with non-negligible probability. We will construct an adversary $\mathcal{B}$ that can break the security of $\mathsf{Core\ 1\text{-}ABE}$ with non-negligible probability. Suppose $\mathcal{C}$ is the challenger for the $\mathsf{Core\ 1\text{-}ABE}$ security game. $\mathcal{B}$ will run $\mathcal{A}$ and simulate the hybrid games.

We describe the reduction $\mathcal{B}$ next. $\mathcal{C}$ samples two messages $\mu^{(0)}, \mu^{(1)} \xleftarrow{\$} \mathbb{Z}_q$, samples a challenge bit $\beta \xleftarrow{\$} \{0,1\}$ and provides $\mu^{(0)}$ to $\mathcal{B}$. $\mathcal{B}$ proceeds as in Hyb_1 except that when it obtains the challenge query $(\mathsf{msg}_0, \mathsf{msg}_1, (\mathbf{M}, \rho), \{\mathsf{pk}_i\}_{i\in U_\mathcal{A}})$ from $\mathcal{A}$, $\mathcal{B}$ makes oracle query for $\mathcal{C}.\mathcal{O}_F((\mathbf{M}, \rho))$. $\mathcal{C}$ computes secret shares of $\mu^{(\beta)}$ as $\{\mu_x\} \leftarrow \mathsf{LSSS.Share}((\mathbf{M}, \rho), \mu^{(\beta)})$ and responds with $(\{\mu_x\}_{\rho(x)=0}, \{\mathsf{CPA.Enc}(\mathbf{u}_{\rho(x)}, \mu_x)\}_{\rho(x)\neq 0})$, where $\mathsf{CPA.Enc}(\mathbf{u}_{\rho(x)}, \mu_x) = ([\![\mathbf{r}_x]\!]_1, [\![\mu_x + \mathbf{r}_x^T \mathbf{u}_{\rho(x)}]\!]_1)$. The way $\mathcal{B}$ embeds this information obtained from $\mathcal{C}$ in its response to $\mathcal{A}$ is as follows: $\mathcal{B}$ samples $\widetilde{\mathbf{c}_x} \xleftarrow{\$} \mathbb{Z}_q^k$ and sets $[\![\mathbf{c}_x]\!]_1 := [\![\mathbf{A}_1^T \widetilde{\mathbf{c}_x}]\!]_1 \cdot (\mathbf{A}_2^T \odot [\![\mathbf{r}_x]\!]_1)$. $\mathcal{B}$ computes $c_0, c_{1,x}, c_{2,x}$ as in Hyb_1. For $c_{3,x}$, $\mathcal{B}$ computes secret shares of $-\mu^{(0)} \mathbf{B}_3^{*T}$ as $\{\widetilde{\omega_x}\} \leftarrow \mathsf{LSSS.Share}((\mathbf{M}, \rho), -\mu^{(0)} \mathbf{B}_3^{*T})$ and then computes $c_{3,x}$ as follows:

$$c_{3,x} := \begin{cases} [\![\widetilde{\omega_x}]\!]_1 \cdot (\mathbf{s}_x^T \odot \mathsf{pk}_{\rho(x),1}) \cdot ([\![\mu_x + \mathbf{r}_x^T \mathbf{u}_{\rho(x)}]\!]_1 \odot \mathbf{B}_3^{*T}) & \text{, if } x \in U_\mathcal{A} \text{ and } \rho(x) \neq 0 \\ [\![\widetilde{\omega_x}]\!]_1 \cdot (\mathbf{s}_x^T \odot \mathsf{pk}_{\rho(x),1}) \cdot ([\![1]\!]_1 \odot \mu_x \mathbf{B}_3^{*T}) & \text{, if } x \in U_\mathcal{A} \text{ and } \rho(x) = 0 \\ [\![\widetilde{\omega_x}]\!]_1 \cdot ([\![\mathbf{c}_x^T]\!]_1 \odot \widetilde{\mathbf{U}_{\rho(x)}}) \cdot ([\![\mu_x + \mathbf{r}_x^T \mathbf{u}_{\rho(x)}]\!]_1 \odot \mathbf{B}_3^{*T}) & \text{, if } x \in \overline{U_\mathcal{A}} \text{ and } \rho(x) \neq 0 \\ [\![\widetilde{\omega_x}]\!]_1 \cdot ([\![\mathbf{c}_x^T]\!]_1 \odot \widetilde{\mathbf{U}_{\rho(x)}}) \cdot ([\![1]\!]_1 \odot \mu_x \mathbf{B}_3^{*T}) & \text{, if } x \in \overline{U_\mathcal{A}} \text{ and } \rho(x) = 0 \end{cases},$$

where for oracle $\mathsf{AuthSetup}(i)$, $\mathcal{B}$ samples master secret key as $\mathsf{msk}_i = (\mathbf{V}_i \xleftarrow{\$} \mathbb{Z}_q^{(2k)\times(2k+1)}, \widetilde{\mathbf{U}_i} \xleftarrow{\$} \mathbb{Z}_q^{(2k)\times(2k+1)})$. $\mathcal{B}$ programs oracles $\mathsf{H}_1(\mathsf{GID})$, $\mathsf{KGen}(i, \mathsf{GID})$, $\mathsf{Corrupt}(i)$ as in Hyb_1. $\mathcal{B}$ implicitly sets

$$\mathbf{U}_i = \widetilde{\mathbf{U}_i} + \mathbf{A}_2^* \mathbf{u}_i \mathbf{B}_3^{*T} \text{ ($\mathcal{B}$ does not know $\mathbf{u}_i$ that is chosen by $\mathcal{C}$).}$$

Observe that when we change $\widetilde{\mathbf{U}_i}$ to $\mathbf{U}_i$, only pk_i, $\mathsf{sk}_{i,\mathsf{GID}}$ and ct are changed. We anaylze the effect of this change next.

- pk_i **remains unchanged:** Since $\mathbf{A}_1\mathbf{A}_2^* = \mathbf{0}$, it follows that $\mathsf{pk}_{i,1} = [\![\mathbf{A}_1\mathbf{U}_i]\!]_1 = [\![\mathbf{A}_1\widetilde{\mathbf{U}_i} + \mathbf{A}_1\mathbf{A}_2^*\mathbf{u}_i\mathbf{B}_3^{*T}]\!]_1 = [\![\mathbf{A}_1\widetilde{\mathbf{U}_i}]\!]_1$.
- $\mathsf{sk}_{i,\mathsf{GID}}$ **remains unchanged for all** GID**:** Since $\mathbf{B}_3^{*T}\mathbf{B}_1 = \mathbf{0}$, it follows that $\mathsf{sk}_{i,\mathsf{GID}} = [\![\mathbf{V}_i\mathbf{k} + \mathbf{U}_i\mathbf{B}_1\mathbf{h}_{\mathsf{GID}}]\!]_2 = [\![\mathbf{V}_i\mathbf{k} + \widetilde{\mathbf{U}_i}\mathbf{B}_1\mathbf{h}_{\mathsf{GID}} + \mathbf{A}_2^*\mathbf{u}_i\mathbf{B}_3^{*T}\mathbf{B}_1\mathbf{h}_{\mathsf{GID}}]\!]_2 = [\![\mathbf{V}_i\mathbf{k} + \widetilde{\mathbf{U}_i}\mathbf{B}_1\mathbf{h}_{\mathsf{GID}}]\!]_2$.
- For the challenge ciphertext ct, observe that the only change is in the $c_{3,x}$ component where $x \in \overline{U_{\mathcal{A}}}$. For x such that $\rho(x) \neq 0$, we have:

$$\begin{aligned}
c_{3,x} &= [\![\widetilde{\omega_x}]\!]_1 \cdot ([\![\mathbf{c}_x^T]\!]_1 \odot \widetilde{\mathbf{U}_{\rho(x)}}) \cdot ([\![\mu_x + \mathbf{r}_x^T\mathbf{u}_{\rho(x)}]\!]_1 \odot \mathbf{B}_3^{*T}) \\
&= [\![\widetilde{\omega_x} + \mathbf{c}_x^T\widetilde{\mathbf{U}_{\rho(x)}} + (\mu_x + \mathbf{r}_x^T\mathbf{u}_{\rho(x)})\mathbf{B}_3^{*T}]\!]_1 \\
&= [\![(\widetilde{\omega_x} + \mu_x\mathbf{B}_3^{*T}) + \mathbf{c}_x^T\widetilde{\mathbf{U}_{\rho(x)}} + \mathbf{r}_x^T\mathbf{u}_{\rho(x)}\mathbf{B}_3^{*T}]\!]_1 \\
&= [\![(\widetilde{\omega_x} + \mu_x\mathbf{B}_3^{*T}) + \mathbf{c}_x^T\mathbf{U}_{\rho(x)} - \mathbf{c}_x^T\mathbf{A}_2^*\mathbf{u}_{\rho(x)}\mathbf{B}_3^{*T} + \mathbf{r}_x^T\mathbf{u}_{\rho(x)}\mathbf{B}_3^{*T}]\!]_1 \\
&= [\![(\widetilde{\omega_x} + \mu_x\mathbf{B}_3^{*T}) + \mathbf{c}_x^T\mathbf{U}_{\rho(x)} - (\mathbf{A}_1^T\widetilde{\mathbf{c}_x} + \mathbf{A}_2^T\mathbf{r}_x)^T\mathbf{A}_2^*\mathbf{u}_{\rho(x)}\mathbf{B}_3^{*T} + \mathbf{r}_x^T\mathbf{u}_{\rho(x)}\mathbf{B}_3^{*T}]\!]_1 \\
&= [\![(\widetilde{\omega_x} + \mu_x\mathbf{B}_3^{*T}) + \mathbf{c}_x^T\mathbf{U}_{\rho(x)}]\!]_1.
\end{aligned}$$

  For x such that $\rho(x) = 0$, we have that $\mathbf{u}_{\rho(x)} = \mathbf{u}_0 = \mathbf{0}$. So, $\mathbf{U}_{\rho(x)} = \widetilde{\mathbf{U}_{\rho(x)}}$. Thus, we have:

$$\begin{aligned}
c_{3,x} &= [\![\widetilde{\omega_x}]\!]_1 \cdot ([\![\mathbf{c}_x^T]\!]_1 \odot \widetilde{\mathbf{U}_{\rho(x)}}) \cdot ([\![1]\!]_1 \odot \mu_x\mathbf{B}_3^{*T}) \\
&= [\![\widetilde{\omega_x} + \mathbf{c}_x^T\widetilde{\mathbf{U}_{\rho(x)}} + \mu_x\mathbf{B}_3^{*T}]\!]_1 \\
&= [\![(\widetilde{\omega_x} + \mu_x\mathbf{B}_3^{*T}) + \mathbf{c}_x^T\mathbf{U}_{\rho(x)}]\!]_1.
\end{aligned}$$

  Recall that $\widetilde{\omega_x}$ is a secret share of $-\mu^{(0)}\mathbf{B}_3^{*T}$ and μ_x is a secret share of $\mu^{(\beta)}$. Then, from linearity of the LSSS secret sharing scheme, it follows that $(\widetilde{\omega_x} + \mu_x\mathbf{B}_3^{*T})$ is a secret share of $-\mu^{(0)}\mathbf{B}_3^{*T} + \mu^{(\beta)}\mathbf{B}_3^{*T} = (\mu^{(\beta)} - \mu^{(0)})\mathbf{B}_3^{*T}$.

Observe that if $\mathcal{A}$ satisfies the game condition, then $\mathcal{B}$ is admissible in its game with $\mathcal{C}$. Observe that when $\mathcal{C}$ chooses $\beta = 0$, $(\widetilde{\omega_x} + \mu_x\mathbf{B}_3^{*T})$ are secret shares of $\mathbf{0}^T$ and thus $\mathcal{B}$ perfectly simulates Hyb_1. Similarly, when $\mathcal{C}$ chooses $\beta = 1$, $(\widetilde{\omega_x} + \mu_x\mathbf{B}_3^{*T})$ are secret shares of $(\mu^{(1)} - \mu^{(0)})\mathbf{B}_3^{*T}$ and thus $\mathcal{B}$ perfectly simulates Hyb_2 where γ is set implicitly as $\gamma = \mu^{(1)} - \mu^{(0)}$. Thus, if $\mathcal{A}$ can distinguish between Hyb_1 and Hyb_2 non-negligible probability, then $\mathcal{B}$ can break the security of Core 1-ABE with non-negligible probability.

□

Claim C.5. *If the game condition holds and* Core 1-ABE *is secure, then,* $\mathsf{Hyb}_{3,j,1}$ *and* $\mathsf{Hyb}'_{3,j,1}$ *are computationally indistinguishable.*

Proof. Observe that the only difference between $\mathsf{Hyb}_{3,j,1}$ and $\mathsf{Hyb}'_{3,j,1}$ is that in ciphertext component $c_{3,x}$ for all $x \in [n]$: $c_{3,x}$ contains $[\![\omega_x]\!]_1$, where ω_x is a secret share of $\gamma\mathbf{B}_3^{*T} \in \mathbb{Z}_q^{1\times(2k+1)}$ in $\mathsf{Hyb}_{3,j,1}$, but it is a secret share of $(\mathbf{B}_2^*\boldsymbol{\delta} + \gamma\mathbf{B}_3^*)^T$ in $\mathsf{Hyb}'_{3,j,1}$. Therefore, to prove that the hybrids are computationally indistinguishable, we will reduce the indistinguishability to the security of Core 1-ABE.

Suppose that there exists an adversary $\mathcal{A}$ that can distinguish between $\mathsf{Hyb}_{3,j,1}$ and $\mathsf{Hyb}'_{3,j,1}$ with non-negligible probability. We will construct an adversary $\mathcal{B}$ that can break the security of Core 1-ABE with non-negligible probability. Suppose $\mathcal{C}$ is the challenger for the Core 1-ABE security game. $\mathcal{B}$ will run $\mathcal{A}$ and simulate the hybrid games.

We describe the reduction $\mathcal{B}$ next. $\mathcal{C}$ samples two messages $\mu^{(0)}, \mu^{(1)} \xleftarrow{\$} \mathbb{Z}_q$, samples a challenge bit $\beta \xleftarrow{\$} \{0,1\}$ and provides $\mu^{(0)}$ to $\mathcal{B}$. $\mathcal{B}$ proceeds as in $\mathsf{Hyb}_{3,j,1}$ except that when it obtains the challenge query $(\mathsf{msg}_0, \mathsf{msg}_1, (\mathbf{M}, \rho), \{\mathsf{pk}_i\}_{i \in U_{\mathcal{A}}})$ from $\mathcal{A}$, $\mathcal{B}$ makes oracle query for $\mathcal{C}.\mathcal{O}_F((\mathbf{M}, \rho))$. $\mathcal{C}$ computes secret shares of $\mu^{(\beta)}$ as $\{\mu_x\} \leftarrow \mathsf{LSSS.Share}((\mathbf{M}, \rho), \mu^{(\beta)})$ and responds with $(\{\mu_x\}_{\rho(x)=0}, \{\mathsf{CPA.Enc}(\mathbf{u}_{\rho(x)}, \mu_x)\}_{\rho(x)\neq 0})$, where $\mathsf{CPA.Enc}(\mathbf{u}_{\rho(x)}, \mu_x) = ([\![\mathbf{r}_x]\!]_1, [\![\mu_x + \mathbf{r}_x^T \mathbf{u}_{\rho(x)}]\!]_1)$. The way $\mathcal{B}$ embeds this information obtained from $\mathcal{C}$ in its response to $\mathcal{A}$ is as follows: $\mathcal{B}$ samples $\widetilde{\mathbf{c}}_x \xleftarrow{\$} \mathbb{Z}_q^k$ and sets $[\![\mathbf{c}_x]\!]_1 := [\![\mathbf{A}_1^T \widetilde{\mathbf{c}}_x]\!]_1 \cdot (\mathbf{A}_2^T \odot [\![\mathbf{r}_x]\!]_1)$. $\mathcal{B}$ computes $c_0, c_{1,x}, c_{2,x}$ as in $\mathsf{Hyb}_{3,j,1}$. For $c_{3,x}$, $\mathcal{B}$ samples $\gamma \xleftarrow{\$} \mathbb{Z}_q$, $\boldsymbol{\delta}_0 \xleftarrow{\$} \mathbb{Z}_q^k$ and computes secret shares of $(-\mu^{(0)} \mathbf{B}_2^* \boldsymbol{\delta}_0 + \gamma \mathbf{B}_3^*)^T$ as $\{\widetilde{\omega_x}\} \leftarrow \mathsf{LSSS.Share}((\mathbf{M}, \rho), (-\mu^{(0)} \mathbf{B}_2^* \boldsymbol{\delta}_0 + \gamma \mathbf{B}_3^*)^T)$ and then computes $c_{3,x}$ as follows:

$$c_{3,x} := \begin{cases} [\![\widetilde{\omega_x}]\!]_1 \cdot (\mathbf{s}_x^T \odot \mathsf{pk}_{\rho(x),1}) \cdot ([\![\mu_x + \mathbf{r}_x^T \mathbf{u}_{\rho(x)}]\!]_1 \odot \boldsymbol{\delta}_0{}^T \mathbf{B}_2^{*T}) & \text{, if } x \in U_{\mathcal{A}} \text{ and } \rho(x) \neq 0 \\ [\![\widetilde{\omega_x}]\!]_1 \cdot (\mathbf{s}_x^T \odot \mathsf{pk}_{\rho(x),1}) \cdot ([\![1]\!]_1 \odot \mu_x \boldsymbol{\delta}_0{}^T \mathbf{B}_2^{*T}) & \text{, if } x \in U_{\mathcal{A}} \text{ and } \rho(x) = 0 \\ [\![\widetilde{\omega_x}]\!]_1 \cdot ([\![\mathbf{c}_x^T]\!]_1 \odot \widetilde{\mathbf{U}_{\rho(x)}}) \cdot ([\![\mu_x + \mathbf{r}_x^T \mathbf{u}_{\rho(x)}]\!]_1 \odot \boldsymbol{\delta}_0{}^T \mathbf{B}_2^{*T}) & \text{, if } x \in \overline{U_{\mathcal{A}}} \text{ and } \rho(x) \neq 0 \\ [\![\widetilde{\omega_x}]\!]_1 \cdot ([\![\mathbf{c}_x^T]\!]_1 \odot \widetilde{\mathbf{U}_{\rho(x)}}) \cdot ([\![1]\!]_1 \odot \mu_x \boldsymbol{\delta}_0{}^T \mathbf{B}_2^{*T}) & \text{, if } x \in \overline{U_{\mathcal{A}}} \text{ and } \rho(x) = 0 \end{cases},$$

where for oracle $\mathsf{AuthSetup}(i)$, $\mathcal{B}$ samples master secret key as $\mathsf{msk}_i = (\mathbf{V}_i \xleftarrow{\$} \mathbb{Z}_q^{(2k)\times(2k+1)}, \widetilde{\mathbf{U}}_i \xleftarrow{\$} \mathbb{Z}_q^{(2k)\times(2k+1)})$. $\mathcal{B}$ programs oracles $\mathsf{H}_1(\mathsf{GID})$, $\mathsf{KGen}(i, \mathsf{GID})$, $\mathsf{Corrupt}(i)$ as in $\mathsf{Hyb}_{3,j,1}$. $\mathcal{B}$ implicitly sets

$$\mathbf{U}_i = \widetilde{\mathbf{U}}_i + \mathbf{A}_2^* \mathbf{u}_i \boldsymbol{\delta}_0{}^T \mathbf{B}_2^{*T} \text{ ($\mathcal{B}$ does not know $\mathbf{u}_i$ that is chosen by $\mathcal{C}$).}$$

Observe that when we change $\widetilde{\mathbf{U}}_i$ to $\mathbf{U}_i$, only pk_i, $\mathsf{sk}_{i,\mathsf{GID}}$ and ct are changed. We anaylze the effect of this change next.

- pk_i **remains unchanged:** Since $\mathbf{A}_1 \mathbf{A}_2^* = \mathbf{0}$, it follows that $\mathsf{pk}_{i,1} = [\![\mathbf{A}_1 \mathbf{U}_i]\!]_1 = [\![\mathbf{A}_1 \widetilde{\mathbf{U}}_i + \mathbf{A}_1 \mathbf{A}_2^* \mathbf{u}_i \boldsymbol{\delta}_0{}^T \mathbf{B}_2^{*T}]\!]_1 = [\![\mathbf{A}_1 \widetilde{\mathbf{U}}_i]\!]_1$.
- $\mathsf{sk}_{i,\mathsf{GID}_{\mathsf{index}}}$ **remains unchanged for all** $\mathsf{GID}_{\mathsf{index}}$ **when** $\mathsf{index} \neq j$**. For** $\mathsf{index} = j$**,** $\mathsf{sk}_{i,\mathsf{GID}_{\mathsf{index}}}$ **changes but $\mathcal{B}$ can simulate the change using oracle query** $\mathcal{C}.\mathcal{O}_X(\{i\})$**.** Recall that $\mathsf{sk}_{i,\mathsf{GID}_{\mathsf{index}}}$ is of the form $[\![\mathbf{V}_i \mathbf{k}]\!]_2 \cdot (\mathbf{U}_i \odot \mathsf{H}_1(\mathsf{GID}_{\mathsf{index}}))$, where $\mathsf{H}_1(\mathsf{GID}_{\mathsf{index}})$ is of the following form depending on index:

$$\mathsf{H}_1(\mathsf{GID}_{\mathsf{index}}) = \begin{cases} [\![\mathbf{B}_1 \mathbf{h}_{\mathsf{GID}_{\mathsf{index}}} + \mathbf{B}_3]\!]_2 & \text{, if } \mathsf{index} \leq j-1 \\ [\![\mathbf{B}_1 \mathbf{h}_{\mathsf{GID}_{\mathsf{index}}} + \mathbf{B}_2 \mathbf{h}'_{\mathsf{GID}_{\mathsf{index}}}]\!]_2 & \text{, if } \mathsf{index} = j \\ [\![\mathbf{B}_1 \mathbf{h}_{\mathsf{GID}_{\mathsf{index}}}]\!]_2 & \text{, if } \mathsf{index} > j \end{cases}.$$

  We analyze the three cases separately:

  - **Case 1:** $\mathsf{index} < j-1$**.** $\mathsf{sk}_{i,\mathsf{GID}_{\mathsf{index}}}$ **remains unchanged.** Since $\mathbf{B}_2^{*T} \mathbf{B}_1 = \mathbf{0}, \mathbf{B}_2^{*T} \mathbf{B}_3 = \mathbf{0}$, it follows that

$$\begin{aligned} \mathsf{sk}_{i,\mathsf{GID}_{\mathsf{index}}} &= [\![\mathbf{V}_i \mathbf{k} + \mathbf{U}_i (\mathbf{B}_1 \mathbf{h}_{\mathsf{GID}_{\mathsf{index}}} + \mathbf{B}_3)]\!]_2 \\ &= [\![\mathbf{V}_i \mathbf{k} + (\widetilde{\mathbf{U}}_i + \mathbf{A}_2^* \mathbf{u}_i \boldsymbol{\delta}_0{}^T \mathbf{B}_2^{*T})(\mathbf{B}_1 \mathbf{h}_{\mathsf{GID}_{\mathsf{index}}} + \mathbf{B}_3)]\!]_2 \\ &= [\![\mathbf{V}_i \mathbf{k} + \widetilde{\mathbf{U}}_i (\mathbf{B}_1 \mathbf{h}_{\mathsf{GID}_{\mathsf{index}}} + \mathbf{B}_3)]\!]_2. \end{aligned}$$

  - **Case 2:** $\mathsf{index} = j$**.** $\mathsf{sk}_{i,\mathsf{GID}_{\mathsf{index}}}$ **is simulated with the help of oracle query** $\mathcal{C}.\mathcal{O}_X(\{i\})$**.** Since $\mathbf{B}_2^{*T} \mathbf{B}_1 = \mathbf{0}, \mathbf{B}_2^{*T} \mathbf{B}_2 = \mathbf{I}$, it follows that

$$\begin{aligned} \mathsf{sk}_{i,\mathsf{GID}_{\mathsf{index}}} &= [\![\mathbf{V}_i \mathbf{k} + \mathbf{U}_i (\mathbf{B}_1 \mathbf{h}_{\mathsf{GID}_{\mathsf{index}}} + \mathbf{B}_2 \mathbf{h}'_{\mathsf{GID}_{\mathsf{index}}})]\!]_2 \\ &= [\![\mathbf{V}_i \mathbf{k} + (\widetilde{\mathbf{U}}_i + \mathbf{A}_2^* \mathbf{u}_i \boldsymbol{\delta}_0{}^T \mathbf{B}_2^{*T})(\mathbf{B}_1 \mathbf{h}_{\mathsf{GID}_{\mathsf{index}}} + \mathbf{B}_2 \mathbf{h}'_{\mathsf{GID}_{\mathsf{index}}})]\!]_2 \\ &= [\![\mathbf{V}_i \mathbf{k} + \widetilde{\mathbf{U}}_i (\mathbf{B}_1 \mathbf{h}_{\mathsf{GID}_{\mathsf{index}}} + \mathbf{B}_2 \mathbf{h}'_{\mathsf{GID}_{\mathsf{index}}}) + \mathbf{A}_2^* \mathbf{u}_i \boldsymbol{\delta}_0{}^T \mathbf{h}'_{\mathsf{GID}_{\mathsf{index}}}]\!]_2. \end{aligned}$$

So, to compute $\mathsf{sk}_{i,\mathsf{GID}_{\mathsf{index}}}$, $\mathcal{B}$ needs to know $\mathbf{u}_i$. For this, $\mathcal{B}$ queries the oracle $\mathcal{C}.\mathcal{O}_X(\{i\})$ and obtains $\mathbf{u}_i$ as a response. Using $\mathbf{u}_i$, $\mathcal{B}$ can compute $\mathbf{U}_i$ and using this, it can properly simulate $\mathsf{sk}_{i,\mathsf{GID}_{\mathsf{index}}}$.

– **Case 3:** $\mathsf{index} > j$**.** $\mathsf{sk}_{i,\mathsf{GID}_{\mathsf{index}}}$ **remains unchanged.** Since $\mathbf{B}_2^{*T}\mathbf{B}_1 = \mathbf{0}$, it follows that

$$\begin{aligned}\mathsf{sk}_{i,\mathsf{GID}_{\mathsf{index}}} &= [\![\mathbf{V}_i\mathbf{k} + \mathbf{U}_i\mathbf{B}_1\mathbf{h}_{\mathsf{GID}_{\mathsf{index}}}]\!]_2\\ &= [\![\mathbf{V}_i\mathbf{k} + (\widetilde{\mathbf{U}_i} + \mathbf{A}_2^*\mathbf{u}_i{\boldsymbol{\delta}_0}^T\mathbf{B}_2^{*T})\mathbf{B}_1\mathbf{h}_{\mathsf{GID}_{\mathsf{index}}}]\!]_2\\ &= [\![\mathbf{V}_i\mathbf{k} + \widetilde{\mathbf{U}_i}\mathbf{B}_1\mathbf{h}_{\mathsf{GID}_{\mathsf{index}}}]\!]_2.\end{aligned}$$

- For the challenge ciphertext ct, observe that the only change is in the $c_{3,x}$ component where $x \in \overline{U_{\mathcal{A}}}$. For x such that $\rho(x) \neq 0$, we have:

$$\begin{aligned}c_{3,x} &= [\![\widetilde{\omega_x}]\!]_1 \cdot ([\![\mathbf{c}_x^T]\!]_1 \odot \widetilde{\mathbf{U}_{\rho(x)}}) \cdot ([\![\mu_x + \mathbf{r}_x^T\mathbf{u}_{\rho(x)}]\!]_1 \odot {\boldsymbol{\delta}_0}^T\mathbf{B}_2^{*T})\\ &= [\![\widetilde{\omega_x} + \mathbf{c}_x^T\widetilde{\mathbf{U}_{\rho(x)}} + (\mu_x + \mathbf{r}_x^T\mathbf{u}_{\rho(x)}){\boldsymbol{\delta}_0}^T\mathbf{B}_2^{*T}]\!]_1\\ &= [\![(\widetilde{\omega_x} + \mu_x{\boldsymbol{\delta}_0}^T\mathbf{B}_2^{*T}) + \mathbf{c}_x^T\widetilde{\mathbf{U}_{\rho(x)}} + \mathbf{r}_x^T\mathbf{u}_{\rho(x)}{\boldsymbol{\delta}_0}^T\mathbf{B}_2^{*T}]\!]_1\\ &= [\![(\widetilde{\omega_x} + \mu_x{\boldsymbol{\delta}_0}^T\mathbf{B}_2^{*T}) + \mathbf{c}_x^T\mathbf{U}_{\rho(x)} - \mathbf{c}_x^T\mathbf{A}_2^*\mathbf{u}_{\rho(x)}{\boldsymbol{\delta}_0}^T\mathbf{B}_2^{*T} + \mathbf{r}_x^T\mathbf{u}_{\rho(x)}{\boldsymbol{\delta}_0}^T\mathbf{B}_2^{*T}]\!]_1\\ &= [\![(\widetilde{\omega_x} + \mu_x{\boldsymbol{\delta}_0}^T\mathbf{B}_2^{*T}) + \mathbf{c}_x^T\mathbf{U}_{\rho(x)} - (\mathbf{A}_1^T\widetilde{\mathbf{c}_x} + \mathbf{A}_2^T\mathbf{r}_x)^T\mathbf{A}_2^*\mathbf{u}_{\rho(x)}{\boldsymbol{\delta}_0}^T\mathbf{B}_2^{*T} + \mathbf{r}_x^T\mathbf{u}_{\rho(x)}{\boldsymbol{\delta}_0}^T\mathbf{B}_2^{*T}]\!]_1\\ &= [\![(\widetilde{\omega_x} + \mu_x{\boldsymbol{\delta}_0}^T\mathbf{B}_2^{*T}) + \mathbf{c}_x^T\mathbf{U}_{\rho(x)}]\!]_1.\end{aligned}$$

  For x such that $\rho(x) = 0$, we have that $\mathbf{u}_{\rho(x)} = \mathbf{u}_0 = \mathbf{0}$. So, $\mathbf{U}_{\rho(x)} = \widetilde{\mathbf{U}_{\rho(x)}}$. Thus, we have:

$$\begin{aligned}c_{3,x} &= [\![\widetilde{\omega_x}]\!]_1 \cdot ([\![\mathbf{c}_x^T]\!]_1 \odot \widetilde{\mathbf{U}_{\rho(x)}}) \cdot ([\![1]\!]_1 \odot \mu_x{\boldsymbol{\delta}_0}^T\mathbf{B}_2^{*T})\\ &= [\![\widetilde{\omega_x} + \mathbf{c}_x^T\widetilde{\mathbf{U}_{\rho(x)}} + \mu_x{\boldsymbol{\delta}_0}^T\mathbf{B}_2^{*T}]\!]_1\\ &= [\![(\widetilde{\omega_x} + \mu_x{\boldsymbol{\delta}_0}^T\mathbf{B}_2^{*T}) + \mathbf{c}_x^T\mathbf{U}_{\rho(x)}]\!]_1.\end{aligned}$$

  Recall that $\widetilde{\omega_x}$ is a secret share of $(-\mu^{(0)}\mathbf{B}_2^*\boldsymbol{\delta}_0 + \gamma\mathbf{B}_3^*)^T$ and μ_x is a secret share of $\mu^{(\beta)}$. Then, from linearity of the LSSS secret sharing scheme, it follows that $(\widetilde{\omega_x} + \mu_x{\boldsymbol{\delta}_0}^T\mathbf{B}_2^{*T})$ is a secret share of $(-\mu^{(0)}\mathbf{B}_2^*\boldsymbol{\delta}_0 + \gamma\mathbf{B}_3^*)^T + \mu^{(\beta)}{\boldsymbol{\delta}_0}^T\mathbf{B}_2^{*T} = (\mu^{(\beta)} - \mu^{(0)}){\boldsymbol{\delta}_0}^T\mathbf{B}_2^{*T} + \gamma\mathbf{B}_3^{*T}$.

Observe that if $\mathcal{A}$ satisfies the game condition, then $\mathcal{B}$ is admissible in its game with $\mathcal{C}$. Observe that when $\mathcal{C}$ chooses $\beta = 0$, $(\widetilde{\omega_x} + \mu_x\mathbf{B}_3^{*T})$ are secret shares of $\gamma\mathbf{B}_3^{*T}$ and thus $\mathcal{B}$ perfectly simulates $\mathsf{Hyb}_{3,j,1}$. Similarly, when $\mathcal{C}$ chooses $\beta = 1$, $(\widetilde{\omega_x} + \mu_x\mathbf{B}_3^{*T})$ are secret shares of $(\mu^{(\beta)} - \mu^{(0)}){\boldsymbol{\delta}_0}^T\mathbf{B}_2^{*T} + \gamma\mathbf{B}_3^{*T}$ and thus $\mathcal{B}$ perfectly simulates $\mathsf{Hyb}'_{3,j,1}$ where $\boldsymbol{\delta}$ is set implicitly as $\boldsymbol{\delta} = (\mu^{(1)} - \mu^{(0)})\boldsymbol{\delta}_0$. Thus, if $\mathcal{A}$ can distinguish between $\mathsf{Hyb}_{3,j,1}$ and $\mathsf{Hyb}'_{3,j,1}$ non-negligible probability, then $\mathcal{B}$ can break the security of Core 1-ABE with non-negligible probability.

□

Claim C.6. *If the game condition holds and* Core 1-ABE *is secure, then,* $\mathsf{Hyb}_{3,j,2}$ *and* $\mathsf{Hyb}'_{3,j,2}$ *are computationally indistinguishable.*

Proof. Similar to the proof of Claim C.4. □

Claim C.7. *If the game condition holds and* Core 1-ABE *is secure, then,* $\mathsf{Hyb}_{3,q}$ *and* Hyb_4 *are computationally indistinguishable.*

Proof. Observe that the only difference between $\mathsf{Hyb}_{3,q}$ and Hyb_4 is in ciphertext components $c_{2,x}$ for all $x \in [n]$. Observe that $c_{2,x}$ contains $[\![\lambda_x]\!]_1$, where λ_x is a secret share of $\mathbf{t}^T \in \mathbb{Z}_q^{1\times(2k+1)}$ in $\mathsf{Hyb}_{3,q}$, but it is a secret share of $(\mathbf{t}+\tau\mathbf{B}_3^*)^T$ in Hyb_4. Therefore, to prove that the hybrids are computationally indistinguishable, we will reduce the indistinguishability to the security of Core 1-ABE.

Suppose that there exists an adversary $\mathcal{A}$ that can distinguish between $\mathsf{Hyb}_{3,q}$ and Hyb_4 with non-negligible probability. We will construct an adversary $\mathcal{B}$ that can break the security of Core 1-ABE with non-negligible probability. Suppose $\mathcal{C}$ is the challenger for the Core 1-ABE security game. $\mathcal{B}$ will run $\mathcal{A}$ and simulate the hybrid games.

We describe the reduction $\mathcal{B}$ next. $\mathcal{C}$ samples two messages $\mu^{(0)}, \mu^{(1)} \xleftarrow{\$} \mathbb{Z}_q$, samples a challenge bit $\beta \xleftarrow{\$} \{0,1\}$ and provides $\mu^{(0)}$ to $\mathcal{B}$. $\mathcal{B}$ proceeds as in $\mathsf{Hyb}_{3,q}$ except that when it obtains the challenge query $(\mathsf{msg}_0, \mathsf{msg}_1, (\mathbf{M},\rho), \{\mathsf{pk}_i\}_{i\in U_\mathcal{A}})$ from $\mathcal{A}$, $\mathcal{B}$ makes oracle query for $\mathcal{C}.\mathcal{O}_F((\mathbf{M},\rho))$. $\mathcal{C}$ computes secret shares of $\mu^{(\beta)}$ as $\{\mu_x\} \leftarrow \mathsf{LSSS.Share}((\mathbf{M},\rho), \mu^{(\beta)})$ and responds with $(\{\mu_x\}_{\rho(x)=0}, \{\mathsf{CPA.Enc}(\mathbf{u}_{\rho(x)}, \mu_x)\}_{\rho(x)\neq 0})$, where $\mathsf{CPA.Enc}(\mathbf{u}_{\rho(x)}, \mu_x) = ([\![\mathbf{r}_x]\!]_1, [\![\mu_x + \mathbf{r}_x^T\mathbf{u}_{\rho(x)}]\!]_1)$. The way $\mathcal{B}$ embeds this information obtained from $\mathcal{C}$ in its response to $\mathcal{A}$ is as follows: $\mathcal{B}$ samples $\widetilde{\mathbf{c}}_x \xleftarrow{\$} \mathbb{Z}_q^k$ and sets $[\![\mathbf{c}_x]\!]_1 := [\![\mathbf{A}_1^T\widetilde{\mathbf{c}}_x]\!]_1 \cdot (\mathbf{A}_2^T \odot [\![\mathbf{r}_x]\!]_1)$. $\mathcal{B}$ computes $c_0, c_{1,x}$ as in $\mathsf{Hyb}_{3,q}$. For $c_{2,x}$, $\mathcal{B}$ computes secret shares of $(\mathbf{t}-\mu^{(0)}\mathbf{B}_3^*)^T$ as $\{\widetilde{\lambda_x}\} \leftarrow \mathsf{LSSS.Share}((\mathbf{M},\rho), (\mathbf{t}-\mu^{(0)}\mathbf{B}_3^*)^T)$ and then computes $c_{2,x}$ as follows:

$$c_{2,x} := \begin{cases} [\![\widetilde{\lambda_x}]\!]_1 \cdot (\mathbf{s}_x^T \odot \mathsf{pk}_{\rho(x),0}) \cdot ([\![\mu_x + \mathbf{r}_x^T\mathbf{u}_{\rho(x)}]\!]_1 \odot \mathbf{B}_3^{*T}) & \text{, if } x \in U_\mathcal{A} \text{ and } \rho(x) \neq 0 \\ [\![\widetilde{\lambda_x}]\!]_1 \cdot (\mathbf{s}_x^T \odot \mathsf{pk}_{\rho(x),0}) \cdot ([\![1]\!]_1 \odot \mu_x\mathbf{B}_3^{*T}) & \text{, if } x \in U_\mathcal{A} \text{ and } \rho(x) = 0 \\ [\![\widetilde{\lambda_x}]\!]_1 \cdot ([\![\mathbf{c}_x^T]\!]_1 \odot \widetilde{\mathbf{V}_{\rho(x)}}) \cdot ([\![\mu_x + \mathbf{r}_x^T\mathbf{u}_{\rho(x)}]\!]_1 \odot \mathbf{B}_3^{*T}) & \text{, if } x \in \overline{U_\mathcal{A}} \text{ and } \rho(x) \neq 0 \\ [\![\widetilde{\lambda_x}]\!]_1 \cdot ([\![\mathbf{c}_x^T]\!]_1 \odot \widetilde{\mathbf{V}_{\rho(x)}}) \cdot ([\![1]\!]_1 \odot \mu_x\mathbf{B}_3^{*T}) & \text{, if } x \in \overline{U_\mathcal{A}} \text{ and } \rho(x) = 0 \end{cases},$$

where for oracle $\mathsf{AuthSetup}(i)$, $\mathcal{B}$ samples master secret key as $\mathsf{msk}_i = (\widetilde{\mathbf{V}}_i \xleftarrow{\$} \mathbb{Z}_q^{(2k)\times(2k+1)}, \widetilde{\mathbf{U}}_i \xleftarrow{\$} \mathbb{Z}_q^{(2k)\times(2k+1)})$. Since the vector $\mathbf{k}$ in glpbal parameters gp is uniform random, $\mathcal{B}$ can equivalently compute it as $\mathbf{k} := \mathbf{B}_1\mathbf{k}_1 + \mathbf{B}_2\mathbf{k}_2 + k_3\mathbf{B}_3$ for uniform random $\mathbf{k}_1 \xleftarrow{\$} \mathbb{Z}_q^k$, $\mathbf{k}_2 \xleftarrow{\$} \mathbb{Z}_q^k$, $k_3 \xleftarrow{\$} \mathbb{Z}_q$. For $c_{3,x}$, $\mathcal{B}$ computes secret shares of $\gamma'\mathbf{B}_3^{*T}$ as $\{\widetilde{\omega_x}\} \leftarrow \mathsf{LSSS.Share}((\mathbf{M},\rho), \gamma'\mathbf{B}_3^{*T})$, where $\gamma' \xleftarrow{\$} \mathbb{Z}_q$ and then computes $c_{3,x}$ as follows:

$$c_{3,x} := \begin{cases} [\![\widetilde{\omega_x}]\!]_1 \cdot (\mathbf{s}_x^T \odot \mathsf{pk}_{\rho(x),1}) / ([\![\mu_x + \mathbf{r}_x^T\mathbf{u}_{\rho(x)}]\!]_1 \odot k_3\mathbf{B}_3^{*T}) & \text{, if } x \in U_\mathcal{A} \text{ and } \rho(x) \neq 0 \\ [\![\widetilde{\omega_x}]\!]_1 \cdot (\mathbf{s}_x^T \odot \mathsf{pk}_{\rho(x),1}) / ([\![1]\!]_1 \odot \mu_x k_3\mathbf{B}_3^{*T}) & \text{, if } x \in U_\mathcal{A} \text{ and } \rho(x) = 0 \\ [\![\widetilde{\omega_x}]\!]_1 \cdot ([\![\mathbf{c}_x^T]\!]_1 \odot \widetilde{\mathbf{U}_{\rho(x)}}) / ([\![\mu_x + \mathbf{r}_x^T\mathbf{u}_{\rho(x)}]\!]_1 \odot k_3\mathbf{B}_3^{*T}) & \text{, if } x \in \overline{U_\mathcal{A}} \text{ and } \rho(x) \neq 0 \\ [\![\widetilde{\omega_x}]\!]_1 \cdot ([\![\mathbf{c}_x^T]\!]_1 \odot \widetilde{\mathbf{U}_{\rho(x)}}) / ([\![1]\!]_1 \odot \mu_x k_3\mathbf{B}_3^{*T}) & \text{, if } x \in \overline{U_\mathcal{A}} \text{ and } \rho(x) = 0 \end{cases},$$

$\mathcal{B}$ programs oracles $\mathsf{H}_1(\mathsf{GID})$, $\mathsf{KGen}(i, \mathsf{GID})$, $\mathsf{Corrupt}(i)$ as in $\mathsf{Hyb}_{3,q}$. $\mathcal{B}$ implicitly sets

$$\mathbf{V}_i = \widetilde{\mathbf{V}}_i + \mathbf{A}_2^*\mathbf{u}_i\mathbf{B}_3^{*T},\ \mathbf{U}_i = \widetilde{\mathbf{U}}_i - k_3\mathbf{A}_2^*\mathbf{u}_i\mathbf{B}_3^{*T}\ (\mathcal{B} \text{ does not know } \mathbf{u}_i \text{ that is chosen by } \mathcal{C}).$$

Observe that when we change $(\widetilde{\mathbf{V}}_i, \widetilde{\mathbf{U}}_i)$ to $(\mathbf{V}_i, \mathbf{U}_i)$, only pk_i, $\mathsf{sk}_{i,\mathsf{GID}}$ and ct are changed. We anaylze the effect of this change next.

- pk_i **remains unchanged:** Since $\mathbf{A}_1\mathbf{A}_2^* = \mathbf{0}$, it follows that $\mathsf{pk}_{i,0} = [\![\mathbf{A}_1\mathbf{V}_i]\!]_1 = [\![\mathbf{A}_1\widetilde{\mathbf{V}}_i + \mathbf{A}_1\mathbf{A}_2^*\mathbf{u}_i\mathbf{B}_3^{*T}]\!]_1 = [\![\mathbf{A}_1\widetilde{\mathbf{V}}_i]\!]_1$. Similarly, $\mathsf{pk}_{i,1} = [\![\mathbf{A}_1\mathbf{U}_i]\!]_1 = [\![\mathbf{A}_1\widetilde{\mathbf{U}}_i - k_3\mathbf{A}_1\mathbf{A}_2^*\mathbf{u}_i\mathbf{B}_3^{*T}]\!]_1 = [\![\mathbf{A}_1\widetilde{\mathbf{U}}_i]\!]_1$.

- $\mathsf{sk}_{i,\mathsf{GID}}$ **remains unchanged for all** GID**:** Since $\mathbf{B}_3^{*T}\mathbf{B}_1 = \mathbf{0}, \mathbf{B}_3^{*T}\mathbf{B}_2 = \mathbf{0}, \mathbf{B}_3^{*T}\mathbf{B}_3 = \mathbf{I}$, it follows that

$$\begin{aligned}
\mathsf{sk}_{i,\mathsf{GID}} &= [\![\mathbf{V}_i\mathbf{k} + \mathbf{U}_i(\mathbf{B}_1\mathbf{h}_{\mathsf{GID}} + \mathbf{B}_3)]\!]_2 \\
&= [\![(\widetilde{\mathbf{V}_i} + \mathbf{A}_2^*\mathbf{u}_i\mathbf{B}_3^{*T})\mathbf{k} + (\widetilde{\mathbf{U}_i} - k_3\mathbf{A}_2^*\mathbf{u}_i\mathbf{B}_3^{*T})(\mathbf{B}_1\mathbf{h}_{\mathsf{GID}} + \mathbf{B}_3)]\!]_2 \\
&= [\![\widetilde{\mathbf{V}_i}\mathbf{k} + \mathbf{A}_2^*\mathbf{u}_i\mathbf{B}_3^{*T}\mathbf{k} + \widetilde{\mathbf{U}_i}(\mathbf{B}_1\mathbf{h}_{\mathsf{GID}} + \mathbf{B}_3) - k_3\mathbf{A}_2^*\mathbf{u}_i]\!]_2 \\
&= [\![\widetilde{\mathbf{V}_i}\mathbf{k} + \mathbf{A}_2^*\mathbf{u}_i\mathbf{B}_3^{*T}(\mathbf{B}_1\mathbf{k}_1 + \mathbf{B}_2\mathbf{k}_2 + k_3\mathbf{B}_3) + \widetilde{\mathbf{U}_i}(\mathbf{B}_1\mathbf{h}_{\mathsf{GID}} + \mathbf{B}_3) - k_3\mathbf{A}_2^*\mathbf{u}_i]\!]_2 \\
&= [\![\widetilde{\mathbf{V}_i}\mathbf{k} + \widetilde{\mathbf{U}_i}(\mathbf{B}_1\mathbf{h}_{\mathsf{GID}} + \mathbf{B}_3)]\!]_2.
\end{aligned}$$

- For the challenge ciphertext ct, observe that the only change is in the $c_{2,x}, c_{3,x}$ components where $x \in \overline{U_{\mathcal{A}}}$. For x such that $\rho(x) \neq 0$, we have:

$$\begin{aligned}
c_{2,x} &= [\![\widetilde{\lambda_x}]\!]_1 \cdot ([\![\mathbf{c}_x^T]\!]_1 \odot \widetilde{\mathbf{V}_{\rho(x)}}) \cdot ([\![\mu_x + \mathbf{r}_x^T\mathbf{u}_{\rho(x)}]\!]_1 \odot \mathbf{B}_3^{*T}) \\
&= [\![\widetilde{\lambda_x} + \mathbf{c}_x^T\widetilde{\mathbf{V}_{\rho(x)}} + (\mu_x + \mathbf{r}_x^T\mathbf{u}_{\rho(x)})\mathbf{B}_3^{*T}]\!]_1 \\
&= [\![(\widetilde{\lambda_x} + \mu_x\mathbf{B}_3^{*T}) + \mathbf{c}_x^T\widetilde{\mathbf{V}_{\rho(x)}} + \mathbf{r}_x^T\mathbf{u}_{\rho(x)}\mathbf{B}_3^{*T}]\!]_1 \\
&= [\![(\widetilde{\lambda_x} + \mu_x\mathbf{B}_3^{*T}) + \mathbf{c}_x^T\mathbf{V}_{\rho(x)} - \mathbf{c}_x^T\mathbf{A}_2^*\mathbf{u}_{\rho(x)}\mathbf{B}_3^{*T} + \mathbf{r}_x^T\mathbf{u}_{\rho(x)}\mathbf{B}_3^{*T}]\!]_1 \\
&= [\![(\widetilde{\lambda_x} + \mu_x\mathbf{B}_3^{*T}) + \mathbf{c}_x^T\mathbf{V}_{\rho(x)} - (\mathbf{A}_1^T\widetilde{\mathbf{c}_x} + \mathbf{A}_2^T\mathbf{r}_x)^T\mathbf{A}_2^*\mathbf{u}_{\rho(x)}\mathbf{B}_3^{*T} + \mathbf{r}_x^T\mathbf{u}_{\rho(x)}\mathbf{B}_3^{*T}]\!]_1 \\
&= [\![(\widetilde{\lambda_x} + \mu_x\mathbf{B}_3^{*T}) + \mathbf{c}_x^T\mathbf{V}_{\rho(x)}]\!]_1,
\end{aligned}$$

$$\begin{aligned}
c_{3,x} &= [\![\widetilde{\omega_x}]\!]_1 \cdot ([\![\mathbf{c}_x^T]\!]_1 \odot \widetilde{\mathbf{U}_{\rho(x)}})/([\![\mu_x + \mathbf{r}_x^T\mathbf{u}_{\rho(x)}]\!]_1 \odot k_3\mathbf{B}_3^{*T}) \\
&= [\![\widetilde{\omega_x} + \mathbf{c}_x^T\widetilde{\mathbf{U}_{\rho(x)}} - (\mu_x + \mathbf{r}_x^T\mathbf{u}_{\rho(x)})k_3\mathbf{B}_3^{*T}]\!]_1 \\
&= [\![(\widetilde{\omega_x} - \mu_x k_3\mathbf{B}_3^{*T}) + \mathbf{c}_x^T\widetilde{\mathbf{U}_{\rho(x)}} - k_3\mathbf{r}_x^T\mathbf{u}_{\rho(x)}\mathbf{B}_3^{*T}]\!]_1 \\
&= [\![(\widetilde{\omega_x} - \mu_x k_3\mathbf{B}_3^{*T}) + \mathbf{c}_x^T\mathbf{U}_{\rho(x)} + k_3\mathbf{c}_x^T\mathbf{A}_2^*\mathbf{u}_{\rho(x)}\mathbf{B}_3^{*T} - k_3\mathbf{r}_x^T\mathbf{u}_{\rho(x)}\mathbf{B}_3^{*T}]\!]_1 \\
&= [\![(\widetilde{\omega_x} - \mu_x k_3\mathbf{B}_3^{*T}) + \mathbf{c}_x^T\mathbf{U}_{\rho(x)} + k_3(\mathbf{A}_1^T\widetilde{\mathbf{c}_x} + \mathbf{A}_2^T\mathbf{r}_x)^T\mathbf{A}_2^*\mathbf{u}_{\rho(x)}\mathbf{B}_3^{*T} - k_3\mathbf{r}_x^T\mathbf{u}_{\rho(x)}\mathbf{B}_3^{*T}]\!]_1 \\
&= [\![(\widetilde{\omega_x} - \mu_x k_3\mathbf{B}_3^{*T}) + \mathbf{c}_x^T\mathbf{U}_{\rho(x)}]\!]_1.
\end{aligned}$$

  For x such that $\rho(x) = 0$, we have that $\mathbf{u}_{\rho(x)} = \mathbf{u}_0 = \mathbf{0}$. So, $\mathbf{V}_{\rho(x)} = \widetilde{\mathbf{V}_{\rho(x)}}$ and $\mathbf{U}_{\rho(x)} = \widetilde{\mathbf{U}_{\rho(x)}}$. Thus, we have:

$$\begin{aligned}
c_{2,x} &= [\![\widetilde{\lambda_x}]\!]_1 \cdot ([\![\mathbf{c}_x^T]\!]_1 \odot \widetilde{\mathbf{V}_{\rho(x)}}) \cdot ([\![1]\!]_1 \odot \mu_x\mathbf{B}_3^{*T}) \\
&= [\![\widetilde{\lambda_x} + \mathbf{c}_x^T\widetilde{\mathbf{V}_{\rho(x)}} + \mu_x\mathbf{B}_3^{*T}]\!]_1 \\
&= [\![(\widetilde{\lambda_x} + \mu_x\mathbf{B}_3^{*T}) + \mathbf{c}_x^T\mathbf{V}_{\rho(x)}]\!]_1.
\end{aligned}$$

$$\begin{aligned}
c_{3,x} &= [\![\widetilde{\omega_x}]\!]_1 \cdot ([\![\mathbf{c}_x^T]\!]_1 \odot \widetilde{\mathbf{U}_{\rho(x)}})/([\![1]\!]_1 \odot \mu_x k_3\mathbf{B}_3^{*T}) \\
&= [\![\widetilde{\omega_x} + \mathbf{c}_x^T\widetilde{\mathbf{U}_{\rho(x)}} - \mu_x k_3\mathbf{B}_3^{*T}]\!]_1 \\
&= [\![(\widetilde{\omega_x} - \mu_x k_3\mathbf{B}_3^{*T}) + \mathbf{c}_x^T\mathbf{U}_{\rho(x)}]\!]_1
\end{aligned}$$

Recall that $\widetilde{\lambda_x}$ is a secret share of $(\mathbf{t} - \mu^{(0)}\mathbf{B}_3^*)^T$ and μ_x is a secret share of $\mu^{(\beta)}$. Then, from linearity of the LSSS secret sharing scheme, it follows that $(\widetilde{\lambda_x} + \mu_x \mathbf{B}_3^{*T})$ is a secret share of $(\mathbf{t} - \mu^{(0)}\mathbf{B}_3^*)^T + \mu^{(\beta)}\mathbf{B}_3^{*T} = \mathbf{t}^T + (\mu^{(\beta)} - \mu^{(0)})\mathbf{B}_3^{*T}$. Recall that $\widetilde{\omega_x}$ is a secret share of $\gamma'\mathbf{B}_3^{*T}$ and μ_x is a secret share of $\mu^{(\beta)}$. Then, from linearity of the LSSS secret sharing scheme, it follows that $(\widetilde{\omega_x} + \mu_x k_3 \mathbf{B}_3^{*T})$ is a secret share of $\gamma'\mathbf{B}_3^{*T} + \mu^{(\beta)}k_3\mathbf{B}_3^{*T} = (\gamma' + k_3\mu^{(\beta)})\mathbf{B}_3^{*T}$.

Observe that if $\mathcal{A}$ satisfies the game condition, then $\mathcal{B}$ is admissible in its game with $\mathcal{C}$. $\mathcal{B}$ implicitly sets $\gamma = \gamma' + k_3\mu^{(\beta)}$ in $\mathcal{A}$'s view. Since γ' and k_3 are chosen to be uniform random, the distribution of γ is also uniform random for both $\beta \in \{0, 1\}$. Observe that when $\mathcal{C}$ chooses $\beta = 0$, $(\widetilde{\lambda_x} + \mu_x \mathbf{B}_3^{*T})$ are secret shares of $\mathbf{t}^T$ and thus $\mathcal{B}$ perfectly simulates $\mathsf{Hyb}_{3,q}$. Similarly, when $\mathcal{C}$ chooses $\beta = 1$, $(\widetilde{\lambda_x} + \mu_x \mathbf{B}_3^{*T})$ are secret shares of $\mathbf{t}^T + (\mu^{(1)} - \mu^{(0)})\mathbf{B}_3^{*T}$ and thus $\mathcal{B}$ perfectly simulates Hyb_4 where τ is set implicitly as $\tau = \mu^{(1)} - \mu^{(0)}$. Thus, if $\mathcal{A}$ can distinguish between $\mathsf{Hyb}_{3,q}$ and Hyb_4 non-negligible probability, then $\mathcal{B}$ can break the security of Core 1-ABE with non-negligible probability.

□

Thus, from Claims 4.2 to 4.4, 4.6, 4.8, 4.10, 4.12, 4.13 and C.4 to C.7 and hybrid argument, it follows that Theorem 5.1 holds.

D MA-ABE for ASP from prime-order groups: Appendix

D.1 Definition

In case of MA-ABE for ASPs, each attribute will not only be associated with an authority, but also denotes a value in $\mathbb{Z}_q$. So, we modify the definition of MA-ABE presented in Appendix A.5 to incorporate this generalization. We note that this modification is just syntactic and natural when considering MA-ABE for ASPs.

Let $\mathcal{AU}$ denote the authority/attribute universe and let each authority control one attribute. We use authority and attribute interchangably. Suppose that an ASP is denoted by a $(\mathbf{M}, \mathbf{N}, \rho)$ of matrices $\mathbf{M}, \mathbf{N} \in \mathbb{Z}_q^{n \times \ell}$ and a labeling function $\rho : [n] \to U$, where $U \subseteq \mathcal{AU}$ denotes a subset of the attribute universe.

The syntax of KGen algorithm is then as follows.

- $\mathsf{sk}_{i,\mathsf{GID},z_i} \leftarrow \mathsf{KGen}(\mathsf{gp}, \mathsf{msk}_i, \mathsf{GID}, z_i)$: The key generation algorithm takes as input the global parameters gp, a master secret key msk_i of an authority $i \in \mathcal{AU}$ a user's global identifier $\mathsf{GID} \in \mathcal{GID}$, and an attribute value $z_i \in \mathbb{Z}_q$. It outputs a secret key $\mathsf{sk}_{i,\mathsf{GID},z_i}$ for the user.

Correctness. An MA-ABE scheme for ASP is said to be correct if for every $\lambda \in \mathbb{N}$, $\mathsf{msg} \in \mathbb{M}$, $\mathsf{GID} \in \mathcal{GID}$, every ASP $(\mathbf{M}, \mathbf{N}, \rho)$ defined on a set $U \subseteq \mathcal{AU}$ of attributes, and every set of attribute values $S = \{z_{\rho(x)}\}_{x \in S_x}$ (where $S_x \subseteq [n]$ denotes a subset of row indices) which satisfy the access structure (i.e., $(1, 0, \ldots, 0) \in span(\{z_{\rho(x)}\mathbf{M}_x + \mathbf{N}_x\}_{x \in S_x})$), it holds that

$$\Pr\left[\mathsf{msg}' = \mathsf{msg} : \begin{array}{c} \mathsf{gp} \leftarrow \mathsf{GlobalSetup}(1^\lambda), \\ \forall i \in U : (\mathsf{pk}_i, \mathsf{msk}_i) \leftarrow \mathsf{AuthSetup}(\mathsf{gp}, i) \\ \forall x \in S_x : \mathsf{sk}_{\rho(x),\mathsf{GID},z_{\rho(x)}} \leftarrow \mathsf{KGen}(\mathsf{gp}, \mathsf{msk}_{\rho(x)}, \mathsf{GID}, z_{\rho(x)}) \\ \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{gp}, \mathsf{msg}, \mathbb{A}, \{\mathsf{pk}_i\}_{i \in U}) \\ \mathsf{msg}' \leftarrow \mathsf{Dec}(\mathsf{gp}, \mathsf{ct}, \{z_{\rho(x)}, \mathsf{sk}_{\rho(x),\mathsf{GID},z_{\rho(x)}}\}_{x \in S_x}) \end{array}\right] = 1.$$

Full Adaptive Security. The security game remains the same as in Appendix A.5 except that for keygen oracle query $\mathsf{KGen}(i, \mathsf{GID}, z_i)$, the reponse is $\mathsf{sk}_{i,\mathsf{GID},z_i} \leftarrow \mathsf{KGen}(\mathsf{gp}, \mathsf{msk}_i, \mathsf{GID}, z_i)$.

Remark D.1. *Building MA-ABE for ASP satisfying full adaptive security seems to have fundamental barriers as pointed out in Section 1. In a little more detail, the fundamental barrier seems that if the challenge policy $(\mathbf{M}, \mathbf{N}, \rho)$ involves a corrupt authority controlling an attribute corresponding to some x-th rows $\mathbf{M}_x, \mathbf{N}_x$, then it can issue more than one attribute values $z_{\rho(x)}$ and $z'_{\rho(x)}$ that are not equal. Then, observe that the span of $z_{\rho(x)}\mathbf{M}_x + \mathbf{N}_x$ and $z'_{\rho(x)}\mathbf{M}_x + \mathbf{N}_x$ contains the vector $\mathbf{M}_x$. Then consider a scenarios where $(1, 0, \ldots, 0)$ is not in span of either of these two vectors and hence each secret key on its own would be insufficient to decrypt, but suppose that $(1, 0, \ldots, 0)$ is in span of $\mathbf{M}_x$. Then, collusion of the two keys should allow to decrypt. This is a fundamental barrier towards building MA-ABE for ASPs with full adaptive security.*

In light of Remark D.1, we consider two weakenings of full adaptive security defined as follows.

Definition D.2 (Full Adaptive Security with Type 1 Restriction)**.** *An MA-ABE scheme for ASPs is said to be full adaptive secure with type 1 restriction if it satisfies full adaptive security subject to following additional admissibility criteria:*

- *no attribute authority appearing in a ciphertext is corrupted (that is, set S defined in Appendix A.5 and Figure 3 is empty),*
- *the adversary queries at most one key per authority and user id pair (i, GID).*

Suppose the challenge access policy $(\mathbf{M}, \mathbf{N}, \rho)$ is defined over a set of attributes $U \subseteq \mathcal{AU}$, that is, $\rho : [n] \to U$. Then the game condition from Appendix A.5 combined with the additional admissibility criteria described above simplifies to the following requirements:

- *for each $\mathsf{GID} \in \mathcal{GID}$, it is required that $S_{\mathsf{GID}} \notin (\mathbf{M}, \mathbf{N}, \rho)$,*
- *for each $\mathsf{GID} \in \mathcal{GID}$, for each $i \in S_{\mathsf{GID}}$, it is required that $|S_{i,\mathsf{GID}}| = 1$,*

where S_{GID} and $S_{i,\mathsf{GID}}$ are defined as follows:

- *for each global identifier $\mathsf{GID} \in \mathcal{GID}$, S_{GID} denotes the subset of U containing attributes i such that the adversary queried a secret key for the tuple $(i, \mathsf{GID}, \cdot)$.*
- *for each global identifier $\mathsf{GID} \in \mathcal{GID}$, for each attribute $i \in S_{\mathsf{GID}}$, $S_{i,\mathsf{GID}}$ denotes the set of attribute values z_i such that the adversary queried a secret key for the tuple (i, GID, z_i).*

We remark that the adversary is still allowed to make corrupt authorities adaptively subject to the above additional admissibility criteria.

Definition D.3 (Full Adaptive Security with Type 2 Restriction)**.** *An MA-ABE scheme for ASPs is said to be full adaptive secure with type 2 restriction if it satisfies full adaptive security subject to following additional admissibility criteria:*

- *decryption requires keys from all authorities appearing in a ciphertext,*
- *either the adversary corrupts no authority appearing in the challenge ciphertext policy or for each GID queried, there exists an honest authority appearing in the challenge ciphertext policy who did not issue any secret key,*
- *the adversary queries at most one key per authority and user id pair (i, GID).*

D.2 Correctness of MA-ABE for ASP construction

For the MA-ABE for ASP construction in Figure 2, observe that d_x can be simplified as follows:

$$\begin{aligned}
d_x &= e(c_{2,x}^{z_{\rho(x)}} \cdot \widehat{c_{2,x}}, [\![\mathbf{k}]\!]_2) \cdot \frac{e(c_{3,x}^{z_{\rho(x)}} \cdot \widehat{c_{3,x}}, \mathsf{H}_1(\mathsf{GID}))}{e(c_{1,x}, \mathsf{sk}_{\rho(x),\mathsf{GID},z_{\rho(x)}})} \\
&= [\![((z_{\rho(x)}\lambda_x + \widehat{\lambda_x}) + \mathbf{s}_x^T\mathbf{A}(z_{\rho(x)}\mathbf{V}_{\rho(x)} + \widehat{\mathbf{V}_{\rho(x)}}))\mathbf{k}]\!]_T \\
&\quad \cdot \frac{[\![((z_{\rho(x)}\omega_x + \widehat{\omega_x}) + \mathbf{s}_x^T\mathbf{A}(z_{\rho(x)}\mathbf{U}_{\rho(x)} + \widehat{\mathbf{U}_{\rho(x)}}))\mathbf{h}_{\mathsf{GID}}]\!]_T}{[\![(\mathbf{s}_x^T\mathbf{A})((z_{\rho(x)}\mathbf{V}_{\rho(x)} + \widehat{\mathbf{V}_{\rho(x)}})\mathbf{k} + (z_{\rho(x)}\mathbf{U}_{\rho(x)} + \widehat{\mathbf{U}_{\rho(x)}})\mathbf{h}_{\mathsf{GID}})]\!]_T} \\
&= [\![(z_{\rho(x)}\lambda_x + \widehat{\lambda_x})\mathbf{k} + (z_{\rho(x)}\omega_x + \widehat{\omega_x})\mathbf{h}_{\mathsf{GID}}]\!]_T \\
&= [\![(z_{\rho(x)}\mathbf{M}_x + \mathbf{N}_x)(\mathbf{T}\mathbf{k} + \mathbf{W}\mathbf{h}_{\mathsf{GID}})]\!]_T.
\end{aligned}$$

Then, d can be simplified as follows:

$$\begin{aligned}
d &= \prod_{x \in S_x} d_x^{w_x} \\
&= [\![\sum_{x \in S_x} w_x(z_{\rho(x)}\mathbf{M}_x + \mathbf{N}_x)(\mathbf{T}\mathbf{k} + \mathbf{W}\mathbf{h}_{\mathsf{GID}})]\!]_T \\
&= [\![(1, 0, \ldots, 0)(\mathbf{T}\mathbf{k} + \mathbf{W}\mathbf{h}_{\mathsf{GID}})]\!]_T \\
&= [\![\mathbf{t}^T\mathbf{k} + \mathbf{0}^T\mathbf{h}_{\mathsf{GID}}]\!]_T \qquad \text{(because first rows of } \mathbf{T} \text{ and } \mathbf{W} \text{ are } \mathbf{t}^T \text{ and } \mathbf{0}^T\text{)} \\
&= [\![\mathbf{t}^T\mathbf{k}]\!]_T.
\end{aligned}$$

Thus, it follows that $c_0/d = \mathsf{msg}$. Thus, correctness holds.

D.3 Proof of Theorem 6.1

We prove full adaptive security with Type 1 restriction of the MA-ABE scheme for ASP presented in Figure 2 subject to the one-use restriction, that is, ρ is injective. We first formally state Theorem 6.1 next.

Theorem D.4. *The MA-ABE construction for ASP in Figure 2 is fully adaptively secure with Type 1 restriction (Definition D.2) if all of the following hold true.*

- *game condition holds and ρ is injective.*
- *k-MDDH assumption holds in groups $\mathbb{G}_1$ and $\mathbb{G}_2$ (Assumption A.8).*
- $\mathsf{SD}^{\mathbb{G}_2}_{\mathbf{B}_1 \to \mathbf{B}_1, \mathbf{B}_2}$ *assumption holds (Assumption A.9).*
- $\mathsf{SD}^{\mathbb{G}_2}_{\mathbf{B}_2 \to \mathbf{B}_2, \mathbf{B}_3}$ *assumption holds (Assumption A.9).*

We prove this via a sequence of hybrid games. This sequence of hybrids closely follows those for the monotone BSP case presented in Theorem 4.1 and Section 4.1.

Suppose the adversary makes q number of queries to the random oracle H_1. Then, the hybrid games are as follows:

$$\mathsf{Hyb}_{\mathsf{Real}}, \mathsf{Hyb}'_{\mathsf{Real}}, \mathsf{Hyb}_1, \mathsf{Hyb}_2, \{\mathsf{Hyb}_{3,j,1}, \mathsf{Hyb}'_{3,j,1}, \mathsf{Hyb}_{3,j,2}, \mathsf{Hyb}'_{3,j,2}, \mathsf{Hyb}_{3,j,3}\}_{j \in [q]}, \mathsf{Hyb}_4, \mathsf{Hyb}_5.$$

Hybrid $\mathsf{Hyb}_{\mathsf{Real}}$. This is the real-world game $\mathsf{MA\text{-}ABE}_{\mathcal{A}}^{\mathsf{fully\text{-}adaptive}}$ with the additional admissibility criteria for full adaptive security with Type 1 restriction as defined in Definition D.2.

Hybrid $\mathsf{Hyb}'_{\mathsf{Real}}$. This is $\mathsf{Hyb}_{\mathsf{Real}}$ except that the challenger computes $\boxed{(\mathsf{gp}, \mathsf{st}) \leftarrow \mathsf{GlobalSetup}^*(1^\lambda)}$ and provides gp to the adversary. Here, $\mathsf{GlobalSetup}^*$ runs the same computation as $\mathsf{GlobalSetup}$ to compute gp and additionally also computes the following:

$$\mathbf{A}_2 \xleftarrow{\$} \mathbb{Z}_q^{1\times(k+1)}, \mathbf{B}_1, \mathbf{B}_2 \xleftarrow{\$} \mathbb{Z}_q^{(2k+1)\times k}, \mathbf{B}_3 \xleftarrow{\$} \mathbb{Z}_q^{(2k+1)\times 1}$$
$$(\mathbf{A}_1^*, \mathbf{A}_2^*) = \begin{pmatrix}\mathbf{A}_1\\ \mathbf{A}_2\end{pmatrix}^{-1}, (\mathbf{B}_1^*, \mathbf{B}_2^*, \mathbf{B}_3^*) = \left((\mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3)^{-1}\right)^T.$$

Let $\mathbf{A} = \begin{pmatrix}\mathbf{A}_1\\ \mathbf{A}_2\end{pmatrix}$, $\mathbf{A}^* = (\mathbf{A}_1^*, \mathbf{A}_2^*)$, $\mathbf{B} = (\mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3)$, $\mathbf{B}^* = (\mathbf{B}_1^*, \mathbf{B}_2^*, \mathbf{B}_3^*)$. Then, $\mathsf{st} = (\mathbf{A}, \mathbf{A}^*, \mathbf{B}, \mathbf{B}^*)$.

Observe that $\forall i, j \in \{1,2\}$: $\mathbf{A}_i\mathbf{A}_j^* = \mathbf{I}$ if $i = j$, and $\mathbf{0}$ if $i \neq j$. Similarly, $\forall i, j \in \{1,2,3\}$: $\mathbf{B}_i^T\mathbf{B}_j^* = \mathbf{I}$ if $i = j$, and $\mathbf{0}$ if $i \neq j$.

Claim D.5. *Hybrids $\mathsf{Hyb}_{\mathsf{Real}}$ and $\mathsf{Hyb}'_{\mathsf{Real}}$ are identically distributed.*

Proof. $\mathsf{Hyb}_{\mathsf{Real}}$ and $\mathsf{Hyb}'_{\mathsf{Real}}$ are identically distributed because the distribution of gp in $\mathsf{GlobalSetup}$ and $\mathsf{GlobalSetup}^*$ is the same. □

Hybrid Hyb_0. This is same as $\mathsf{Hyb}_{\mathsf{Real}}$ except that the hash function H_1 is programmed to output all hash values in $\mathsf{span}(\mathbf{B}_1)$ as follows: on input GID, sample $\mathbf{h}_{\mathsf{GID}} \xleftarrow{\$} \boxed{\mathbb{Z}_q^k}$ and output $\mathsf{H}_1(\mathsf{GID}) = [\![\boxed{\mathbf{B}_1\mathbf{h}_{\mathsf{GID}}}]\!]_2$.

Claim D.6. *If k-MDDH holds in group $\mathbb{G}_2$, then $\mathsf{Hyb}'_{\mathsf{Real}}$ and Hyb_0 are computationally indistinguishable.*

Proof. Similar to proof of Claim 4.3. □

Hybrid Hyb_1. This is same as $\mathsf{Hyb}'_{\mathsf{Real}}$ except that the ciphertext is changed to semi-functional form. Before presenting the semi-functional form, we note that the normal form of challenge ciphertext can be simply written as follows since no attribute authority appearing in the ciphetext can be corrupted (Definition D.2): $\mathsf{ct} := (c_0, \{c_{1,x}, c_{2,x}, c_{3,x}, \widehat{c_{2,x}}, \widehat{c_{3,x}}\}_{x\in[n]})$, where $c_0 := \mathsf{msg}_b \cdot [\![\mathbf{t}^T\mathbf{k}]\!]_T$, $\forall x \in [n]$:

$$\begin{aligned}
c_{1,x} &:= [\![\mathbf{c}_x^T]\!]_1 := [\![\mathbf{s}_x^T\mathbf{A}_1]\!]_1,\\
c_{2,x} &:= [\![\lambda_x]\!]_1 \cdot (\boxed{\mathbf{c}_x^T \odot [\![\mathbf{V}_{\rho(x)}]\!]_1}),\\
c_{3,x} &:= [\![\omega_x]\!]_1 \cdot (\boxed{\mathbf{c}_x^T \odot [\![\mathbf{U}_{\rho(x)}]\!]_1}),\\
\widehat{c_{2,x}} &:= [\![\widehat{\lambda_x}]\!]_1 \cdot (\boxed{\mathbf{c}_x^T \odot [\![\widehat{\mathbf{V}_{\rho(x)}}]\!]_1}),\\
\widehat{c_{3,x}} &:= [\![\widehat{\omega_x}]\!]_1 \cdot (\boxed{\mathbf{c}_x^T \odot [\![\widehat{\mathbf{U}_{\rho(x)}}]\!]_1}).
\end{aligned}$$

Given this notation, the ciphertext in Hyb_1 is same as the normal ciphertext except that for all $x \in \overline{U_{\mathcal{A}}}$: $\boxed{c_{1,x} := [\![\mathbf{c}_x^T]\!]_1}$, where $\boxed{\mathbf{c}_x \xleftarrow{\$} \mathbb{Z}_q^{k+1}}$. We call this semi-functional ciphertext. Observe that in total this means that semi-functional ciphertext changes $c_{1,x}, c_{2,x}, c_{3,x}, \widehat{c_{2,x}}, \widehat{c_{3,x}}$ for $x \in [n]$ when compared to normal ciphertext.

Claim D.7. *If k-MDDH holds in group $\mathbb{G}_1$, then* Hyb_0 *and* Hyb_1 *are computationally indistinguishable.*

Proof. Similar to the proof of Claim 4.4 □

Hybrid Hyb_2. This is same as Hyb_1 except that the ciphertext structure is changed as follows: for all $x \in [n]$, $c_{3,x}, \widehat{c_{3,x}}$ are computed using $\omega_x := \mathbf{M}_x \boxed{\mathbf{W}}$, $\widehat{\omega_x} := \mathbf{N}_x \boxed{\mathbf{W}}$, where $\mathbf{W}$ is same as in Hyb_1 except that the first row is changed from $\mathbf{0}^T$ to $\gamma \mathbf{B}_3^{*T}$, where $\gamma \xleftarrow{\$} \mathbb{Z}_q$, that is, $\mathbf{W} = \begin{pmatrix} \boxed{\gamma \mathbf{B}_3^{*T}} \\ \mathbf{W}_{\mathsf{bot}} \end{pmatrix}$.

Claim D.8. *If the game condition holds and ρ is injective, then,* Hyb_1 *and* Hyb_2 *are statistically indistinguishable.*

Proof. Observe that the only difference between Hyb_1 and Hyb_2 is that in ciphertext components $c_{3,x}, \widehat{c_{3,x}}$ for all $x \in [n]$: $c_{3,x}, \widehat{c_{3,x}}$ contain $[\![\omega_x]\!]_1, [\![\widehat{\omega_x}]\!]_1$ respectively, where $\omega_x, \widehat{\omega_x}$ are secret share of $\mathbf{0}^T \in \mathbb{Z}_q^{1\times(2k+1)}$ in Hyb_1, but are secret share of $\gamma \mathbf{B}_3^{*T}$ in Hyb_2. Therefore, to prove that the hybrids are statistically indistinguishable, we will argue that $\gamma \mathbf{B}_3^{*T}$ is information theoretically hidden from the adversary $\mathcal{A}$ in Hyb_2.

Since the adversary is not allowed to corrupt any attribute authorities appearing in the challenge ciphertext, the only possible way for $\mathcal{A}$ to get information about $\gamma \mathbf{B}_3^{*T}$ is through the ciphertext components $c_{3,x}, \widehat{c_{3,x}}$ corresponding to all the rows $x \in [n]$ of $\mathbf{N}$. However, for each such row x, $\mathcal{A}$ can only recover $\mathbf{c}_x^T$, $\mathbf{M}_x\mathbf{W} + \mathbf{c}_x^T \mathbf{U}_{\rho(x)}$, $\mathbf{N}_x\mathbf{W} + \mathbf{c}_x^T \widehat{\mathbf{U}_{\rho(x)}}$ information theoretically. Without loss of generality, we can compute $\mathbf{U}_{\rho(x)} := \mathbf{U}_{\rho(x),1}\mathbf{B}_1^{*T} + \mathbf{U}_{\rho(x),2}\mathbf{B}_2^{*T} + \mathbf{U}_{\rho(x),3}\mathbf{B}_3^{*T}$, where $\mathbf{U}_{\rho(x),1} \xleftarrow{\$} \mathbb{Z}_q^{(k+1)\times k}, \mathbf{U}_{\rho(x),2} \xleftarrow{\$} \mathbb{Z}_q^{(k+1)\times k}$, and $\mathbf{U}_{\rho(x),3} \xleftarrow{\$} \mathbb{Z}_q^{(k+1)\times 1}$. Similarly, we can compute $\widehat{\mathbf{U}_{\rho(x)}} := \widehat{\mathbf{U}_{\rho(x),1}}\mathbf{B}_1^{*T} + \widehat{\mathbf{U}_{\rho(x),2}}\mathbf{B}_2^{*T} + \widehat{\mathbf{U}_{\rho(x),3}}\mathbf{B}_3^{*T}$, where $\widehat{\mathbf{U}_{\rho(x),1}} \xleftarrow{\$} \mathbb{Z}_q^{(k+1)\times k}, \widehat{\mathbf{U}_{\rho(x),2}} \xleftarrow{\$} \mathbb{Z}_q^{(k+1)\times k}$, and $\widehat{\mathbf{U}_{\rho(x),3}} \xleftarrow{\$} \mathbb{Z}_q^{(k+1)\times 1}$. Let the first entry of $\mathbf{M}_x$ be m_x, that is, $\mathbf{M}_x = (m_x, \ldots)$. Let the first entry of $\mathbf{N}_x$ be n_x, that is, $\mathbf{N}_x = (n_x, \ldots)$. Then, observe that we can write

$$
\begin{aligned}
&\mathbf{N}_x \begin{pmatrix} \gamma \mathbf{B}_3^{*T} \\ \mathbf{W}_{\mathsf{bot}} \end{pmatrix} + \mathbf{c}_x^T \widehat{\mathbf{U}_{\rho(x)}} \\
&= \mathbf{N}_x \begin{pmatrix} \mathbf{0} \\ \mathbf{W}_{\mathsf{bot}} \end{pmatrix} + \mathbf{N}_x \begin{pmatrix} \gamma \mathbf{B}_3^{*T} \\ \mathbf{0} \end{pmatrix} + \mathbf{c}_x^T \widehat{\mathbf{U}_{\rho(x)}} \\
&= \mathbf{N}_x \begin{pmatrix} \mathbf{0} \\ \mathbf{W}_{\mathsf{bot}} \end{pmatrix} + (n_x, \ldots) \begin{pmatrix} \gamma \mathbf{B}_3^{*T} \\ \mathbf{0} \end{pmatrix} + \mathbf{c}_x^T \widehat{\mathbf{U}_{\rho(x)}} \\
&= \mathbf{N}_x \begin{pmatrix} \mathbf{0} \\ \mathbf{W}_{\mathsf{bot}} \end{pmatrix} + n_x \gamma \mathbf{B}_3^{*T} + \mathbf{c}_x^T \widehat{\mathbf{U}_{\rho(x)}} \\
&= \mathbf{N}_x \begin{pmatrix} \mathbf{0} \\ \mathbf{W}_{\mathsf{bot}} \end{pmatrix} + (n_x \gamma + \mathbf{c}_x^T \widehat{\mathbf{U}_{\rho(x),3}})\mathbf{B}_3^{*T} + \mathbf{c}_x^T \widehat{\mathbf{U}_{\rho(x),1}} \mathbf{B}_1^{*T} + \mathbf{c}_x^T \widehat{\mathbf{U}_{\rho(x),2}} \mathbf{B}_2^{*T} \\
&\equiv \mathbf{N}_x \begin{pmatrix} \mathbf{0} \\ \mathbf{W}_{\mathsf{bot}} \end{pmatrix} + (\mathbf{c}_x^T \widehat{\mathbf{U}'_{\rho(x),3}})\mathbf{B}_3^{*T} + \mathbf{c}_x^T \widehat{\mathbf{U}_{\rho(x),1}} \mathbf{B}_1^{*T} + \mathbf{c}_x^T \widehat{\mathbf{U}_{\rho(x),2}} \mathbf{B}_2^{*T}
\end{aligned}
$$

where we can write $\widehat{\mathbf{U}'_{\rho(x),3}} = \widehat{\mathbf{U}_{\rho(x),3}} + \widehat{\Delta}$ such that $n_x \gamma = \mathbf{c}_x^T \widehat{\Delta}$. Similarly, we can write

$$
\begin{aligned}
&\mathbf{M}_x \begin{pmatrix} \gamma \mathbf{B}_3^{*T} \\ \mathbf{W}_{\mathsf{bot}} \end{pmatrix} + \mathbf{c}_x^T \mathbf{U}_{\rho(x)} \\
&\equiv \mathbf{M}_x \begin{pmatrix} \mathbf{0} \\ \mathbf{W}_{\mathsf{bot}} \end{pmatrix} + (\mathbf{c}_x^T \mathbf{U}'_{\rho(x),3})\mathbf{B}_3^{*T} + \mathbf{c}_x^T \mathbf{U}_{\rho(x),1} \mathbf{B}_1^{*T} + \mathbf{c}_x^T \mathbf{U}_{\rho(x),2} \mathbf{B}_2^{*T}
\end{aligned}
$$

where we can write $\mathbf{U}'_{\rho(x),3} = \mathbf{U}_{\rho(x),3} + \Delta$ such that $m_x\gamma = \mathbf{c}_x^T\Delta$. Therefore, to complete the proof, it suffices to argue that $(\mathbf{U}_{\rho(x),3}, \widehat{\mathbf{U}_{\rho(x),3}})$ and $(\mathbf{U}'_{\rho(x),3}, \widehat{\mathbf{U}'_{\rho(x),3}})$ are identically distributed. We show this next.

Observe that since ρ is injective, hence it follows that $\mathbf{U}_{\rho(x)}, \widehat{\mathbf{U}_{\rho(x)}}$ are fresh random matrices and the only other place they appear are in the secret key $\mathsf{sk}_{\rho(x),\mathsf{GID},z_{\rho(x)}}$. Due to the game condition that $|S_{\rho(x),\mathsf{GID}}| = 1$, there can be only one such secret key per attribute $\rho(x)$. We argue that $\mathsf{sk}_{\rho(x),\mathsf{GID},z_{\rho(x)}}$ information theoretically leaks no information about $(\mathbf{U}_{\rho(x),3}, \widehat{\mathbf{U}_{\rho(x),3}})$ and hence $(\mathbf{U}_{\rho(x),3}, \widehat{\mathbf{U}_{\rho(x),3}})$ and $(\mathbf{U}'_{\rho(x),3}, \widehat{\mathbf{U}'_{\rho(x),3}})$ are identically distributed. To see this, observe that $\mathsf{sk}_{\rho(x),\mathsf{GID},z_{\rho(x)}}$ information theoretically reveals $(z_{\rho(x)} \cdot \mathbf{V}_{\rho(x)} + \widehat{\mathbf{V}_{\rho(x)}})\mathbf{k} + (z_{\rho(x)} \cdot \mathbf{U}_{\rho(x)} + \widehat{\mathbf{U}_{\rho(x)}})\mathbf{B}_1\mathbf{h}_{\mathsf{GID}}$, where $\mathbf{U}_{\rho(x)}\mathbf{B}_1\mathbf{h}_{\mathsf{GID}} = \mathbf{U}_{\rho(x),1}\mathbf{h}_{\mathsf{GID}}$ and $\widehat{\mathbf{U}_{\rho(x)}}\mathbf{B}_1\mathbf{h}_{\mathsf{GID}} = \widehat{\mathbf{U}_{\rho(x),1}}\mathbf{h}_{\mathsf{GID}}$ since $\mathbf{B}_1^{*T}\mathbf{B}_1 = \mathbf{I}$, $\mathbf{B}_2^{*T}\mathbf{B}_1 = \mathbf{0}$ and $\mathbf{B}_3^{*T}\mathbf{B}_1 = \mathbf{0}$, thus no information about $(\mathbf{U}_{\rho(x),3}, \widehat{\mathbf{U}_{\rho(x),3}})$ is revealed.

In conclusion, substituting $(\mathbf{U}_{\rho(x),3}, \widehat{\mathbf{U}_{\rho(x),3}})$ with $(\mathbf{U}'_{\rho(x),3}, \widehat{\mathbf{U}'_{\rho(x),3}})$ (as described above) for all rows $x \in [n]$ of matrices $\mathbf{M}, \mathbf{N}$ allows us to move from Hyb_2 to Hyb_1.

Therefore, it follows that Hyb_1 and Hyb_2 are statistically indistinguishable. This completes the proof of Claim D.8. □

Hybrid $\mathsf{Hyb}_{3,j-1}$ for $j \in [q+1]$. This hybrid is same as Hyb_2 except that for the i^{th} global identifier GID_i for $i \leq j-1$, the challenger programs the output $\mathsf{H}_1(\mathsf{GID}_i)$ of the random oracle H_1 as $\mathsf{H}_1(\mathsf{GID}_i) = [\![\boxed{\mathbf{B}_1\mathbf{h}_{\mathsf{GID}_i} + \mathbf{B}_3}]\!]_2$, where $\mathbf{h}_{\mathsf{GID}_i} \xleftarrow{\$} \mathbb{Z}_q^k$, while for $i > j-1$, it programs the output $\mathsf{H}_1(\mathsf{GID}_i)$ of the random oracle H_1 as $\mathsf{H}_1(\mathsf{GID}_i) = [\![\mathbf{B}_1\mathbf{h}_{\mathsf{GID}_i}]\!]_2$ as earlier.

Observe that $\mathsf{Hyb}_{3,0}$ is same as Hyb_2. We introduce a sequnce of intermediate hybrids $\mathsf{Hyb}_{3,j,1}$, $\mathsf{Hyb}'_{3,j,1}, \mathsf{Hyb}_{3,j,2}, \mathsf{Hyb}'_{3,j,2}, \mathsf{Hyb}_{3,j,3}$ between $\mathsf{Hyb}_{3,j-1}$ and $\mathsf{Hyb}_{3,j}$ for all $j \in [q]$ as defined below.

Hybrid $\mathsf{Hyb}_{3,j,1}$ for $j \in [q]$. This hybrid is same as $\mathsf{Hyb}_{3,j-1}$ except that for the j^{th} global identifier GID_j, the challenger programs the output $\mathsf{H}_1(\mathsf{GID}_j)$ of the random oracle H_1 as $\mathsf{H}_1(\mathsf{GID}_j) = [\![\boxed{\mathbf{B}_1\mathbf{h}_{\mathsf{GID}_j} + \mathbf{B}_2\mathbf{h}'_{\mathsf{GID}_j}}]\!]_2$, where $\mathbf{h}_{\mathsf{GID}_j}, \mathbf{h}'_{\mathsf{GID}_j} \xleftarrow{\$} \mathbb{Z}_q^k$.

Claim D.9. *If $\mathsf{SD}^{\mathbb{G}_2}_{\mathbf{B}_1 \to \mathbf{B}_1,\mathbf{B}_2}$ assumption holds (Assumption A.9), then, hybrids $\mathsf{Hyb}_{3,j-1}$ and $\mathsf{Hyb}_{3,j,1}$ are computationally indistinguishable for all $j \in [q]$.*

Proof. Similar to proof of Claim 4.6. □

Hybrid $\mathsf{Hyb}'_{3,j,1}$. This is same as $\mathsf{Hyb}_{3,j,1}$ except that the ciphertext structure is changed as follows: for all $x \in [n]$, $c_{3,x}, \widehat{c_{3,x}}$ are computed using $\omega_x := \mathbf{M}_x\boxed{\mathbf{W}}$, $\widehat{\omega_x} := \mathbf{N}_x\boxed{\mathbf{W}}$, where $\mathbf{W}$ is same as in Hyb_2 except that the first row is changed from $\gamma\mathbf{B}_3^{*T}$ to $(\mathbf{B}_2^*\boldsymbol{\delta} + \gamma\mathbf{B}_3^*)^T$, where $\gamma \xleftarrow{\$} \mathbb{Z}_q$ and $\boldsymbol{\delta} \xleftarrow{\$} \mathbb{Z}_q^k$ that is, $\mathbf{W} = \begin{pmatrix} \boxed{(\mathbf{B}_2^*\boldsymbol{\delta} + \gamma\mathbf{B}_3^*)^T} \\ \hline \mathbf{W}_{\mathsf{bot}} \end{pmatrix}$.

Claim D.10. *If the game condition holds and ρ is injective, then, $\mathsf{Hyb}_{3,j,1}$ and $\mathsf{Hyb}'_{3,j,1}$ are statistically indistinguishable.*

Proof. Observe that the only difference between $\mathsf{Hyb}_{3,j,1}$ and $\mathsf{Hyb}'_{3,j,1}$ is that in ciphertext components $c_{3,x}, \widehat{c_{3,x}}$ for all $x \in [n]$: $c_{3,x}, \widehat{c_{3,x}}$ contain $[\![\omega_x]\!]_1, [\![\widehat{\omega_x}]\!]_1$ respectively, where $\omega_x, \widehat{\omega_x}$ are secret share of $\gamma\mathbf{B}_3^{*T} \in \mathbb{Z}_q^{1\times(2k+1)}$ in $\mathsf{Hyb}_{3,j,1}$, but it is a secret share of $(\mathbf{B}_2^*\boldsymbol{\delta} + \gamma\mathbf{B}_3^*)^T$ in $\mathsf{Hyb}'_{3,j,1}$. Therefore, to

prove that the hybrids are statistically indistinguishable, we will argue that $\mathbf{B}_2^*\boldsymbol{\delta}$ is information theoretically hidden to the adversary $\mathcal{A}$ in $\mathsf{Hyb}'_{3,j,1}$.

Suppose the challenge access policy $(\mathbf{M}, \mathbf{N}, \rho)$ is defined over a set of attributes $U \subseteq \mathcal{AU}$, that is, $\rho : [n] \to U$. Recall from Definition D.2 that the game condition requires that:

- for each $\mathsf{GID} \in \mathcal{GID}$, it is required that $S_{\mathsf{GID}} \notin (\mathbf{M}, \mathbf{N}, \rho)$,
- for each $\mathsf{GID} \in \mathcal{GID}$, for each $i \in S_{\mathsf{GID}}$, it is required that $|S_{i,\mathsf{GID}}| = 1$,

where S_{GID} and $S_{i,\mathsf{GID}}$ are defined as follows:

- for each global identifier $\mathsf{GID} \in \mathcal{GID}$, S_{GID} denotes the subset of U containing attributes i such that the adversary queried a secret key for the tuple $(i, \mathsf{GID}, \cdot)$.
- for each global identifier $\mathsf{GID} \in \mathcal{GID}$, for each attribute $i \in S_{\mathsf{GID}}$, $S_{i,\mathsf{GID}}$ denotes the set of attribute values z_i such that the adversary queried a secret key for the tuple (i, GID, z_i).

To show that $\mathbf{B}_2^*\boldsymbol{\delta}$ is information theoretically hidden from the adversary $\mathcal{A}$ in $\mathsf{Hyb}'_{3,j,1}$, we only need to rely on the two game conditions for the j^{th} GID, that is, we will use the fact that $S_{\mathsf{GID}_j} \notin (\mathbf{M}, \mathbf{N}, \rho)$ and for each $i \in S_{\mathsf{GID}_j}$, $|S_{i,\mathsf{GID}_j}| = 1$. Here, $S_{\mathsf{GID}_j} \notin (\mathbf{M}, \mathbf{N}, \rho)$ is a shorthand for $(1, 0, \ldots, 0) \notin \mathsf{rowSpan}(\{z_{\rho(x)}\mathbf{M}_x + \mathbf{N}_x\}_{\rho(x) \in S_{\mathsf{GID}_j}})$.

Since the adversary is not allowed to corrupt any attribute authorities appearing in the challenge ciphertext, the only possible way for $\mathcal{A}$ to get information about $\mathbf{B}_2^*\boldsymbol{\delta}$ is through the ciphertext components $c_{3,x}, \widehat{c_{3,x}}$ corresponding to all the rows $x \in [n]$ of $\mathbf{M}, \mathbf{N}$. However, for each such row x, $\mathcal{A}$ can only recover $\mathbf{c}_x^T$, $\mathbf{M}_x\mathbf{W} + \mathbf{c}_x^T\mathbf{U}_{\rho(x)}$, $\mathbf{N}_x\mathbf{W} + \mathbf{c}_x^T\widehat{\mathbf{U}_{\rho(x)}}$ information theoretically. We analyze two cases:

- x **such that** $\rho(x) \in S_{\mathsf{GID}_j}$**:** These are the rows of $\mathbf{M}, \mathbf{N}$ labeled by authorities who issued key for GID_j. From the game condition, it follows that $(1, 0, \ldots, 0) \notin \mathsf{rowSpan}(\{z_{\rho(x)}\mathbf{M}_x + \mathbf{N}_x\}_{\rho(x) \in S_{\mathsf{GID}_j}})$. Therefore, it follows that $\mathbf{M}_x\mathbf{W}$ and $\mathbf{N}_x\mathbf{W}$ contain no information about $(\mathbf{B}_2^*\boldsymbol{\delta} + \gamma\mathbf{B}_3^*)^T$. Hence, $\mathbf{c}_x^T$, $\mathbf{M}_x\mathbf{W} + \mathbf{c}_x^T\mathbf{U}_{\rho(x)}$, $\mathbf{N}_x\mathbf{W} + \mathbf{c}_x^T\widehat{\mathbf{U}_{\rho(x)}}$ reveal no information about $(\mathbf{B}_2^*\boldsymbol{\delta} + \gamma\mathbf{B}_3^*)^T$ to the adversary $\mathcal{A}$.
- x **such that** $\rho(x) \in U \setminus S_{\mathsf{GID}_j}$**:** These are the rows of $\mathbf{M}, \mathbf{N}$ labeled by authorities who are neither corrupt nor issued key for GID_j. The game condition does not apply for these rows and the analysis requires more care as described next. Without loss of generality, we can compute $\mathbf{U}_{\rho(x)} := \mathbf{U}_{\rho(x),1}\mathbf{B}_1^{*T} + \mathbf{U}_{\rho(x),2}\mathbf{B}_2^{*T} + \mathbf{U}_{\rho(x),3}\mathbf{B}_3^{*T}$, where $\mathbf{U}_{\rho(x),1} \xleftarrow{\$} \mathbb{Z}_q^{(k+1)\times k}$, $\mathbf{U}_{\rho(x),2} \xleftarrow{\$} \mathbb{Z}_q^{(k+1)\times k}$, and $\mathbf{U}_{\rho(x),3} \xleftarrow{\$} \mathbb{Z}_q^{(k+1)\times 1}$. Similarly, we can compute $\widehat{\mathbf{U}_{\rho(x)}} := \widehat{\mathbf{U}_{\rho(x),1}}\mathbf{B}_1^{*T} + \widehat{\mathbf{U}_{\rho(x),2}}\mathbf{B}_2^{*T} + \widehat{\mathbf{U}_{\rho(x),3}}\mathbf{B}_3^{*T}$, where $\widehat{\mathbf{U}_{\rho(x),1}} \xleftarrow{\$} \mathbb{Z}_q^{(k+1)\times k}$, $\widehat{\mathbf{U}_{\rho(x),2}} \xleftarrow{\$} \mathbb{Z}_q^{(k+1)\times k}$, and $\widehat{\mathbf{U}_{\rho(x),3}} \xleftarrow{\$} \mathbb{Z}_q^{(k+1)\times 1}$. Let the first entry of $\mathbf{M}_x$ be m_x, that is, $\mathbf{M}_x = (m_x, \ldots)$. Let the first entry of $\mathbf{N}_x$ be n_x, that is,

$\mathbf{N}_x = (n_x, \ldots)$. Then, observe that we can write

$$\begin{aligned}
&\mathbf{N}_x \begin{pmatrix} (\mathbf{B}_2^* \boldsymbol{\delta} + \gamma \mathbf{B}_3^*)^T \\ \mathbf{W}_{\mathsf{bot}} \end{pmatrix} + \mathbf{c}_x^T \widehat{\mathbf{U}_{\rho(x)}} \\
&= \mathbf{N}_x \begin{pmatrix} \gamma \mathbf{B}_3^{*T} \\ \mathbf{W}_{\mathsf{bot}} \end{pmatrix} + \mathbf{N}_x \begin{pmatrix} (\mathbf{B}_2^* \boldsymbol{\delta})^T \\ \mathbf{0} \end{pmatrix} + \mathbf{c}_x^T \widehat{\mathbf{U}_{\rho(x)}} \\
&= \mathbf{N}_x \begin{pmatrix} \gamma \mathbf{B}_3^{*T} \\ \mathbf{W}_{\mathsf{bot}} \end{pmatrix} + (n_x, \ldots) \begin{pmatrix} (\mathbf{B}_2^* \boldsymbol{\delta})^T \\ \mathbf{0} \end{pmatrix} + \mathbf{c}_x^T \widehat{\mathbf{U}_{\rho(x)}} \\
&= \mathbf{N}_x \begin{pmatrix} \gamma \mathbf{B}_3^{*T} \\ \mathbf{W}_{\mathsf{bot}} \end{pmatrix} + n_x (\mathbf{B}_2^* \boldsymbol{\delta})^T + \mathbf{c}_x^T \widehat{\mathbf{U}_{\rho(x)}} \\
&= \mathbf{N}_x \begin{pmatrix} \gamma \mathbf{B}_3^{*T} \\ \mathbf{W}_{\mathsf{bot}} \end{pmatrix} + (n_x \boldsymbol{\delta}^T + \mathbf{c}_x^T \widehat{\mathbf{U}_{\rho(x),2}}) \mathbf{B}_2^{*T} + \mathbf{c}_x^T \widehat{\mathbf{U}_{\rho(x),1}} \mathbf{B}_1^{*T} + \mathbf{c}_x^T \widehat{\mathbf{U}_{\rho(x),3}} \mathbf{B}_3^{*T} \\
&\equiv \mathbf{N}_x \begin{pmatrix} \gamma \mathbf{B}_3^{*T} \\ \mathbf{W}_{\mathsf{bot}} \end{pmatrix} + (\mathbf{c}_x^T \widehat{\mathbf{U}'_{\rho(x),2}}) \mathbf{B}_2^{*T} + \mathbf{c}_x^T \widehat{\mathbf{U}_{\rho(x),1}} \mathbf{B}_1^{*T} + \mathbf{c}_x^T \widehat{\mathbf{U}_{\rho(x),3}} \mathbf{B}_3^{*T}
\end{aligned}$$

where we can write $\widehat{\mathbf{U}'_{\rho(x),2}} = \widehat{\mathbf{U}_{\rho(x),2}} + \widehat{\Delta}$ such that $n_x \boldsymbol{\delta}^T = \mathbf{c}_x^T \widehat{\Delta}$. Similarly, we can write

$$\begin{aligned}
&\mathbf{M}_x \begin{pmatrix} (\mathbf{B}_2^* \boldsymbol{\delta} + \gamma \mathbf{B}_3^*)^T \\ \mathbf{W}_{\mathsf{bot}} \end{pmatrix} + \mathbf{c}_x^T \mathbf{U}_{\rho(x)} \\
&\equiv \mathbf{N}_x \begin{pmatrix} \gamma \mathbf{B}_3^{*T} \\ \mathbf{W}_{\mathsf{bot}} \end{pmatrix} + (\mathbf{c}_x^T \mathbf{U}'_{\rho(x),2}) \mathbf{B}_2^{*T} + \mathbf{c}_x^T \mathbf{U}_{\rho(x),1} \mathbf{B}_1^{*T} + \mathbf{c}_x^T \mathbf{U}_{\rho(x),3} \mathbf{B}_3^{*T}
\end{aligned}$$

where we can write $\mathbf{U}'_{\rho(x),2} = \mathbf{U}_{\rho(x),2} + \Delta$ such that $m_x \boldsymbol{\delta}^T = \mathbf{c}_x^T \Delta$. Therefore, to complete the proof, it suffices to argue that $(\mathbf{U}_{\rho(x),2}, \widehat{\mathbf{U}_{\rho(x),2}})$ and $(\mathbf{U}'_{\rho(x),2}, \widehat{\mathbf{U}'_{\rho(x),2}})$ are identically distributed. We show this next.

Observe that since ρ is injective, hence it follows that $\mathbf{U}_{\rho(x)}, \widehat{\mathbf{U}_{\rho(x)}}$ are fresh random matrix and the only other place it appears are in secret key $\mathsf{sk}_{\rho(x),\mathsf{GID}_{\mathsf{index}}}$. Due to the game condition that $|S_{\rho(x),\mathsf{GID}}| = 1$, there can be only one such secret key per attribute $\rho(x)$. We argue that $\mathsf{sk}_{\rho(x),\mathsf{GID},z_{\rho(x)}}$ information theoretically leaks no information about $(\mathbf{U}_{\rho(x)}, \widehat{\mathbf{U}_{\rho(x)}})$ and hence $(\mathbf{U}_{\rho(x)}, \widehat{\mathbf{U}_{\rho(x)}})$ and $(\mathbf{U}'_{\rho(x)}, \widehat{\mathbf{U}'_{\rho(x)}})$ are identically distributed. To see this, observe that the $\mathbf{U}_{\rho(x)}, \widehat{\mathbf{U}_{\rho(x)}}$-dependent term of $\mathsf{sk}_{\rho(x),\mathsf{GID}_{\mathsf{index}},z_{\rho(x)}}$ is of the form $(z_{\rho(x)} \mathbf{U}_{\rho(x)} + \widehat{\mathbf{U}_{\rho(x)}}) \odot \mathsf{H}_1(\mathsf{GID}_{\mathsf{index}})$, where $\mathsf{H}_1(\mathsf{GID}_{\mathsf{index}})$ is of the following form depending on index:

$$\mathsf{H}_1(\mathsf{GID}_{\mathsf{index}}) = \begin{cases} [\![\mathbf{B}_1 \mathbf{h}_{\mathsf{GID}_{\mathsf{index}}} + \mathbf{B}_3]\!]_2 & , \text{if } \mathsf{index} \leq j-1 \\ [\![\mathbf{B}_1 \mathbf{h}_{\mathsf{GID}_{\mathsf{index}}} + \mathbf{B}_2 \mathbf{h}'_{\mathsf{GID}_{\mathsf{index}}}]\!]_2 & , \text{if } \mathsf{index} = j \\ [\![\mathbf{B}_1 \mathbf{h}_{\mathsf{GID}_{\mathsf{index}}}]\!]_2 & , \text{if } \mathsf{index} > j \end{cases} .$$

We analyze the three cases separately:

– **Case 1:** $\mathsf{index} < j$**.** Observe that the $\mathbf{U}_{\rho(x)}, \widehat{\mathbf{U}_{\rho(x)}}$-dependent term of $\mathsf{sk}_{\rho(x),\mathsf{GID}_{\mathsf{index}},z_{\rho(x)}}$ information theoretically reveals

$$(z_{\rho(x)} \mathbf{U}_{\rho(x),1} + \widehat{\mathbf{U}_{\rho(x),1}}) \mathbf{h}_{\mathsf{GID}_{\mathsf{index}}} + z_{\rho(x)} \mathbf{U}_{\rho(x),3} + \widehat{\mathbf{U}_{\rho(x),3}}$$

since $\mathbf{B}_2^{*T} \mathbf{B}_1 = \mathbf{0}$ and $\mathbf{B}_2^{*T} \mathbf{B}_3 = \mathbf{0}$.

– **Case 2:** $\mathsf{index} = j$**.** This case requires no analysis since adversary $\mathcal{A}$ never sees secret keys $\mathsf{sk}_{\rho(x),\mathsf{GID}_j}$. This is due to the definition of set S_{GID_j} and the fact that we are only considering x such that $\rho(x) \in U \setminus S_{\mathsf{GID}_j}$.

– **Case 3:** $\mathsf{index} > j$**.** Observe that the $\mathbf{U}_{\rho(x)}, \widehat{\mathbf{U}_{\rho(x)}}$-dependent term of $\mathsf{sk}_{\rho(x),\mathsf{GID}_{\mathsf{index}},z_{\rho(x)}}$ information theoretically reveals

$$(z_{\rho(x)}\mathbf{U}_{\rho(x),1} + \widehat{\mathbf{U}_{\rho(x),1}})\mathbf{h}_{\mathsf{GID}_{\mathsf{index}}}$$

since $\mathbf{B}_2^{*T}\mathbf{B}_1 = \mathbf{0}$.

Hence, it follows that $\mathsf{sk}_{\rho(x),\mathsf{GID}_{\mathsf{index}},z_{\rho(x)}}$ information theoretically reveals no information about $(\mathbf{U}_{\rho(x),2}, \widehat{\mathbf{U}_{\rho(x),2}})$, so $(\mathbf{U}_{\rho(x),2}, \widehat{\mathbf{U}_{\rho(x),2}})$ and $(\mathbf{U}'_{\rho(x),2}, \widehat{\mathbf{U}'_{\rho(x),2}})$ are identically distributed. Thus, for all x such that $\rho(x) \in U \setminus S_{\mathsf{GID}_j}$, ciphertext components $c_{2,x}, \widehat{c_{2,x}}$ reveal no information about $\mathbf{B}_2^*\boldsymbol{\delta}$ to the adversary $\mathcal{A}$.

In conclusion, substituting $(\mathbf{U}_{\rho(x),2}, \widehat{\mathbf{U}_{\rho(x),2}})$ with $(\mathbf{U}'_{\rho(x),2}, \widehat{\mathbf{U}'_{\rho(x),2}})$ (as described above) for all rows $x \in [n]$ of matrices $\mathbf{M}, \mathbf{N}$ allows us to move from $\mathsf{Hyb}'_{3,j,1}$ to $\mathsf{Hyb}_{3,j,1}$.

Therefore, it follows that $\mathsf{Hyb}_{3,j,1}$ and $\mathsf{Hyb}'_{3,j,1}$ are statistically indistinguishable. This completes the proof of Claim D.10.

□

Hybrid $\mathsf{Hyb}_{3,j,2}$**.** This is same as $\mathsf{Hyb}'_{3,j,1}$ except that for the j^{th} global identifier GID_j, the challenger programs the output $\mathsf{H}_1(\mathsf{GID}_j)$ of the random oracle H_1 as $\mathsf{H}_1(\mathsf{GID}_j) = [\![\boxed{\mathbf{B}_1\mathbf{h}_{\mathsf{GID}_j} + \mathbf{B}_2\mathbf{h}'_{\mathsf{GID}_j} + \mathbf{B}_3}]\!]_2$, where $\mathbf{h}_{\mathsf{GID}_j}, \mathbf{h}'_{\mathsf{GID}_j} \xleftarrow{\$} \mathbb{Z}_q^k$.

Claim D.11. *If* $\mathsf{SD}^{\mathbb{G}_2}_{\mathbf{B}_2 \to \mathbf{B}_2,\mathbf{B}_3}$ *assumption holds (Assumption A.9), then, hybrids* $\mathsf{Hyb}'_{3,j,1}$ *and* $\mathsf{Hyb}_{3,j,2}$ *are computationally indistinguishable for all* $j \in [q]$.

Proof. Similar to proof of Claim 4.8. □

Hybrid $\mathsf{Hyb}'_{3,j,2}$**.** This is same as $\mathsf{Hyb}_{3,j,2}$ except that the ciphertext structure is changed as follows: the first row of matrix $\mathbf{W}$ is changed from $(\mathbf{B}_2^*\boldsymbol{\delta} + \gamma\mathbf{B}_3^*)^T$ to $\gamma\mathbf{B}_3^{*T}$, where and $\gamma \xleftarrow{\$} \mathbb{Z}_q$, that is, $\mathbf{W} = \begin{pmatrix} \boxed{\gamma\mathbf{B}_3^{*T}} \\ \hline \mathbf{W}_{\mathsf{bot}} \end{pmatrix}$.

Claim D.12. *If the game condition holds and* ρ *is injective, then,* $\mathsf{Hyb}_{3,j,2}$ *and* $\mathsf{Hyb}'_{3,j,2}$ *are statistically indistinguishable.*

Proof. Similar to the proof of Claim D.8. □

Claim D.13. *If* $\mathsf{SD}^{\mathbb{G}_2}_{\mathbf{B}_1 \to \mathbf{B}_1,\mathbf{B}_2}$ *assumption holds (Assumption A.9), then, hybrids* $\mathsf{Hyb}'_{3,j,2}$ *and* $\mathsf{Hyb}_{3,j}$ *are computationally indistinguishable for all* $j \in [q]$.

Proof. Similar to proof of Claim D.9. □

Hybrid Hyb_4. This is same as hybrid $\mathsf{Hyb}_{3,q}$ except that the ciphertext structure is changed as follows: for all $x \in [n]$, $c_{2,x}, \widehat{c_{2,x}}$ are computed using $\lambda_x := \mathbf{M}_x \boxed{\mathbf{T}}$, $\widehat{\lambda_x} := \mathbf{N}_x \boxed{\mathbf{T}}$, where $\mathbf{T}$ is same as $\mathbf{T}$ except that the first row is changed from $\mathbf{t}^T$ to $(\mathbf{t} + \tau \mathbf{B}_3^*)^T$, where $\tau \xleftarrow{\$} \mathbb{Z}_q$, that is $\mathbf{T} = \begin{pmatrix} \boxed{(\mathbf{t} + \tau \mathbf{B}_3^*)^T} \\ \mathbf{T}_{\mathsf{bot}} \end{pmatrix}$. We note that c_0 remains unchanged, that is, the masking term is still $\mathbf{t}^T \mathbf{k}$.

Claim D.14. *If the game condition holds and ρ is injective, then, $\mathsf{Hyb}_{3,q}$ and Hyb_4 are statistically indistinguishable.*

Proof. Observe that the only difference between $\mathsf{Hyb}_{3,q}$ and Hyb_4 is in ciphertext components $c_{2,x}, \widehat{c_{2,x}}$ for all $x \in [n]$. Observe that $c_{2,x}, \widehat{c_{2,x}}$ contains $\lambda_x, [\![\widehat{\lambda_x}]\!]_1$ respectively, where $\lambda_x, \widehat{\lambda_x}$ are secret share of $\mathbf{t}^T \in \mathbb{Z}_q^{1 \times (2k+1)}$ in $\mathsf{Hyb}_{3,q}$, but are secret share of $(\mathbf{t} + \tau \mathbf{B}_3^*)^T$ in Hyb_4. Therefore, to prove that the hybrids are statistically indistinguishable, we will argue that $\tau \mathbf{B}_3^{*T}$ is information theoretically hidden to the adversary $\mathcal{A}$ in Hyb_4.

Since the adversary is not allowed to corrupt any attribute authorities appearing in the challenge ciphertext, the only possible way for $\mathcal{A}$ to get information about $\tau \mathbf{B}_3^{*T}$ is through the ciphertext components $c_{2,x}, \widehat{c_{2,x}}$ corresponding to all the rows $x \in [n]$ of $\mathbf{N}$. However, for each such row x, $\mathcal{A}$ can only recover $\mathbf{c}_x^T$, $\mathbf{M}_x \mathbf{T} + \mathbf{c}_x^T \mathbf{V}_{\rho(x)}$, $\mathbf{N}_x \mathbf{T} + \mathbf{c}_x^T \widehat{\mathbf{V}_{\rho(x)}}$ information theoretically. Without loss of generality, we can compute $\mathbf{V}_{\rho(x)} := \mathbf{V}_{\rho(x),1} \mathbf{B}_1^{*T} + \mathbf{V}_{\rho(x),2} \mathbf{B}_2^{*T} + \mathbf{V}_{\rho(x),3} \mathbf{B}_3^{*T}$, where $\mathbf{V}_{\rho(x),1} \xleftarrow{\$} \mathbb{Z}_q^{(k+1) \times k}, \mathbf{V}_{\rho(x),2} \xleftarrow{\$} \mathbb{Z}_q^{(k+1) \times k}$, and $\mathbf{V}_{\rho(x),3} \xleftarrow{\$} \mathbb{Z}_q^{(k+1) \times 1}$. Similarly, we can compute $\widehat{\mathbf{V}_{\rho(x)}} := \widehat{\mathbf{V}_{\rho(x),1}} \mathbf{B}_1^{*T} + \widehat{\mathbf{V}_{\rho(x),2}} \mathbf{B}_2^{*T} + \widehat{\mathbf{V}_{\rho(x),3}} \mathbf{B}_3^{*T}$, where $\widehat{\mathbf{V}_{\rho(x),1}} \xleftarrow{\$} \mathbb{Z}_q^{(k+1) \times k}, \widehat{\mathbf{V}_{\rho(x),2}} \xleftarrow{\$} \mathbb{Z}_q^{(k+1) \times k}$, and $\widehat{\mathbf{V}_{\rho(x),3}} \xleftarrow{\$} \mathbb{Z}_q^{(k+1) \times 1}$. Let the first entry of $\mathbf{M}_x$ be m_x, that is, $\mathbf{M}_x = (m_x, \ldots)$. Let the first entry of $\mathbf{N}_x$ be n_x, that is, $\mathbf{N}_x = (n_x, \ldots)$. Then, observe that we can write

$$
\begin{aligned}
&\mathbf{N}_x \begin{pmatrix} (\mathbf{t} + \tau \mathbf{B}_3^*)^T \\ \mathbf{T}_{\mathsf{bot}} \end{pmatrix} + \mathbf{c}_x^T \widehat{\mathbf{V}_{\rho(x)}} \\
&= \mathbf{N}_x \begin{pmatrix} \mathbf{t}^T \\ \mathbf{T}_{\mathsf{bot}} \end{pmatrix} + \mathbf{N}_x \begin{pmatrix} \tau \mathbf{B}_3^{*T} \\ \mathbf{0} \end{pmatrix} + \mathbf{c}_x^T \widehat{\mathbf{V}_{\rho(x)}} \\
&= \mathbf{N}_x \begin{pmatrix} \mathbf{t}^T \\ \mathbf{T}_{\mathsf{bot}} \end{pmatrix} + (n_x, \ldots) \begin{pmatrix} \tau \mathbf{B}_3^{*T} \\ \mathbf{0} \end{pmatrix} + \mathbf{c}_x^T \widehat{\mathbf{V}_{\rho(x)}} \\
&= \mathbf{N}_x \begin{pmatrix} \mathbf{t}^T \\ \mathbf{T}_{\mathsf{bot}} \end{pmatrix} + n_x \tau \mathbf{B}_3^{*T} + \mathbf{c}_x^T \widehat{\mathbf{V}_{\rho(x)}} \\
&= \mathbf{N}_x \begin{pmatrix} \mathbf{t}^T \\ \mathbf{T}_{\mathsf{bot}} \end{pmatrix} + (n_x \tau + \mathbf{c}_x^T \widehat{\mathbf{V}_{\rho(x),3}}) \mathbf{B}_3^{*T} + \mathbf{c}_x^T \widehat{\mathbf{V}_{\rho(x),1}} \mathbf{B}_1^{*T} + \mathbf{c}_x^T \widehat{\mathbf{V}_{\rho(x),2}} \mathbf{B}_2^{*T} \\
&\equiv \mathbf{N}_x \begin{pmatrix} \mathbf{t}^T \\ \mathbf{T}_{\mathsf{bot}} \end{pmatrix} + (\mathbf{c}_x^T \widehat{\mathbf{V}'_{\rho(x),3}}) \mathbf{B}_3^{*T} + \mathbf{c}_x^T \widehat{\mathbf{V}_{\rho(x),1}} \mathbf{B}_1^{*T} + \mathbf{c}_x^T \widehat{\mathbf{V}_{\rho(x),2}} \mathbf{B}_2^{*T}
\end{aligned}
$$

where we can write $\widehat{\mathbf{V}'_{\rho(x),3}} = \widehat{\mathbf{V}_{\rho(x),3}} + \widehat{\Delta}$ such that $n_x \tau = \mathbf{c}_x^T \widehat{\Delta}$. Similarly, we can write

$$
\begin{aligned}
&\mathbf{M}_x \begin{pmatrix} (\mathbf{t} + \tau \mathbf{B}_3^*)^T \\ \mathbf{T}_{\mathsf{bot}} \end{pmatrix} + \mathbf{c}_x^T \mathbf{V}_{\rho(x)} \\
&\equiv \mathbf{M}_x \begin{pmatrix} \mathbf{t}^T \\ \mathbf{T}_{\mathsf{bot}} \end{pmatrix} + (\mathbf{c}_x^T \mathbf{V}'_{\rho(x),3}) \mathbf{B}_3^{*T} + \mathbf{c}_x^T \mathbf{V}_{\rho(x),1} \mathbf{B}_1^{*T} + \mathbf{c}_x^T \mathbf{V}_{\rho(x),2} \mathbf{B}_2^{*T}
\end{aligned}
$$

where we can write $\mathbf{V}'_{\rho(x),3} = \mathbf{V}_{\rho(x),3} + \Delta$ such that $m_x\tau = \mathbf{c}_x^T\Delta$. Therefore, to complete the proof, it suffices to argue that $(\mathbf{V}_{\rho(x),3}, \widehat{\mathbf{V}_{\rho(x),3}})$ and $(\mathbf{V}'_{\rho(x),3}, \widehat{\mathbf{V}'_{\rho(x),3}})$ are identically distributed. We show this next.

Observe that since ρ is injective, hence it follows that $\mathbf{V}_{\rho(x)}, \widehat{\mathbf{V}_{\rho(x)}}$ are fresh random matrices and the only other place they appear is in secret key $\mathsf{sk}_{\rho(x),\mathsf{GID},z_{\rho(x)}}$. Due to the game condition that $|S_{\rho(x),\mathsf{GID}}| = 1$, there can be only one such secret key per attribute $\rho(x)$. Specifically, $\mathsf{sk}_{\rho(x),\mathsf{GID}}$ information theoretically reveals $(z_{\rho(x)}\mathbf{V}_{\rho(x)} + \widehat{\mathbf{V}_{\rho(x)}})\mathbf{k} + (z_{\rho(x)}\mathbf{U}_{\rho(x)} + \widehat{\mathbf{U}_{\rho(x)}})(\mathbf{B}_1\mathbf{h}_{\mathsf{GID}} + \mathbf{B}_3)$.

Since $\mathbf{k}$ is uniform random, we can equivalently write it as $\mathbf{k} := \mathbf{B}_1\mathbf{k}_1 + \mathbf{B}_2\mathbf{k}_2 + k_3\mathbf{B}_3$ for uniform random $\mathbf{k}_1 \xleftarrow{\$} \mathbb{Z}_q^k$, $\mathbf{k}_2 \xleftarrow{\$} \mathbb{Z}_q^k$, $k_3 \xleftarrow{\$} \mathbb{Z}_q$. Further, we can write $\mathbf{U}_{\rho(x)} := \mathbf{U}_{\rho(x),1}\mathbf{B}_1^{*T} + \mathbf{U}_{\rho(x),2}\mathbf{B}_2^{*T} + \mathbf{U}_{\rho(x),3}\mathbf{B}_3^{*T}$, where $\mathbf{U}_{\rho(x),1} \xleftarrow{\$} \mathbb{Z}_q^{(k+1)\times k}, \mathbf{U}_{\rho(x),2} \xleftarrow{\$} \mathbb{Z}_q^{(k+1)\times k}$, and $\mathbf{U}_{\rho(x),3} \xleftarrow{\$} \mathbb{Z}_q^{(k+1)\times 1}$. Similarly, we can write $\widehat{\mathbf{U}_{\rho(x)}} := \widehat{\mathbf{U}_{\rho(x),1}}\mathbf{B}_1^{*T} + \widehat{\mathbf{U}_{\rho(x),2}}\mathbf{B}_2^{*T} + \widehat{\mathbf{U}_{\rho(x),3}}\mathbf{B}_3^{*T}$, where $\widehat{\mathbf{U}_{\rho(x),1}} \xleftarrow{\$} \mathbb{Z}_q^{(k+1)\times k}, \widehat{\mathbf{U}_{\rho(x),2}} \xleftarrow{\$} \mathbb{Z}_q^{(k+1)\times k}$, and $\widehat{\mathbf{U}_{\rho(x),3}} \xleftarrow{\$} \mathbb{Z}_q^{(k+1)\times 1}$. Then observe that we can write

$$
\begin{aligned}
& z_{\rho(x)}\left(\mathbf{V}_{\rho(x)}\mathbf{k} + \mathbf{U}_{\rho(x)}\mathbf{B}_1\mathbf{h}_{\mathsf{GID}} + \mathbf{U}_{\rho(x)}\mathbf{B}_3\right) \\
& \quad + \widehat{\mathbf{V}_{\rho(x)}}\mathbf{k} + \widehat{\mathbf{U}_{\rho(x)}}\mathbf{B}_1\mathbf{h}_{\mathsf{GID}} + \widehat{\mathbf{U}_{\rho(x)}}\mathbf{B}_3 \\
& = z_{\rho(x)}\left(\mathbf{V}_{\rho(x),1}\mathbf{k}_1 + \mathbf{V}_{\rho(x),2}\mathbf{k}_2 + k_3\mathbf{V}_{\rho(x),3} + \mathbf{U}_{\rho(x),1}\mathbf{h}_{\mathsf{GID}} + \mathbf{U}_{\rho(x),3}\right) \\
& \quad + \widehat{\mathbf{V}_{\rho(x),1}}\mathbf{k}_1 + \widehat{\mathbf{V}_{\rho(x),2}}\mathbf{k}_2 + k_3\widehat{\mathbf{V}_{\rho(x),3}} + \widehat{\mathbf{U}_{\rho(x),1}}\mathbf{h}_{\mathsf{GID}} + \widehat{\mathbf{U}_{\rho(x),3}} \\
& = z_{\rho(x)}\left(\mathbf{V}_{\rho(x),1}\mathbf{k}_1 + \mathbf{V}_{\rho(x),2}\mathbf{k}_2 + k_3\boxed{\mathbf{V}'_{\rho(x),3}} + \mathbf{U}_{\rho(x),1}\mathbf{h}_{\mathsf{GID}} + \boxed{\mathbf{U}_{\rho(x),3}}\right) \\
& \quad + \widehat{\mathbf{V}_{\rho(x),1}}\mathbf{k}_1 + \widehat{\mathbf{V}_{\rho(x),2}}\mathbf{k}_2 + k_3\boxed{\widehat{\mathbf{V}'_{\rho(x),3}}} + \widehat{\mathbf{U}_{\rho(x),1}}\mathbf{h}_{\mathsf{GID}} + \boxed{\widehat{\mathbf{U}'_{\rho(x),3}}}
\end{aligned}
$$

where we can write $\mathbf{V}'_{\rho(x),3} = \mathbf{V}_{\rho(x),3} + \Delta$, $\mathbf{U}'_{\rho(x),3} = \mathbf{U}_{\rho(x),3} - k_3\Delta$, $\widehat{\mathbf{V}'_{\rho(x),3}} = \widehat{\mathbf{V}_{\rho(x),3}} + \widehat{\Delta}$, and $\widehat{\mathbf{U}'_{\rho(x),3}} = \widehat{\mathbf{U}_{\rho(x),3}} - k_3\widehat{\Delta}$, where $\Delta, \widehat{\Delta}$ are as defined above, that is, choose Δ such that $m_x\tau = \mathbf{c}_x^T\Delta$ and $\widehat{\Delta}$ such that $n_x\tau = \mathbf{c}_x^T\widehat{\Delta}$. Therefore, $(\mathbf{V}_{\rho(x),3}, \widehat{\mathbf{V}_{\rho(x),3}})$ and $(\mathbf{V}'_{\rho(x),3}, \widehat{\mathbf{V}'_{\rho(x),3}})$ are identically distributed as long as $(\mathbf{U}_{\rho(x),3}, \widehat{\mathbf{U}_{\rho(x),3}})$ and $(\mathbf{U}'_{\rho(x),3}, \widehat{\mathbf{U}'_{\rho(x),3}})$ are identically distributed. We show this next. Observe that since ρ is injective, hence it follows that $\mathbf{U}_{\rho(x)}, \widehat{\mathbf{U}_{\rho(x)}}$ are fresh random matrix and other than $\mathsf{sk}_{\rho(x),\mathsf{GID},z_{\rho(x)}}$, the only other place they appears are in ciphertext components $c_{3,x}, \widehat{c_{3,x}}$. For ciphertext components $c_{3,x}, \widehat{c_{3,x}}$ corresponding to all the rows of $\mathbf{M}, \mathbf{N}$, $\mathcal{A}$ can recover

$\mathbf{M}_x\mathbf{W} + \mathbf{c}_x^T\mathbf{U}_{\rho(x)}$, $\mathbf{N}_x\mathbf{W} + \mathbf{c}_x^T\widehat{\mathbf{U}_{\rho(x)}}$ information theoretically. Observe that we can write

$$
\begin{aligned}
&\mathbf{N}_x\mathbf{W} + \mathbf{c}_x^T\widehat{\mathbf{U}_{\rho(x)}} \\
&\quad= \mathbf{N}_x\begin{pmatrix}\gamma\mathbf{B}_3^{*T}\\ \mathbf{W}_{\mathsf{bot}}\end{pmatrix} + \mathbf{c}_x^T\widehat{\mathbf{U}_{\rho(x)}} \\
&\quad= \mathbf{N}_x\begin{pmatrix}\gamma\mathbf{B}_3^{*T}\\ \mathbf{W}_{\mathsf{bot}}\end{pmatrix} + \mathbf{c}_x^T\widehat{\mathbf{U}_{\rho(x),1}}\mathbf{B}_1^{*T} + \mathbf{c}_x^T\widehat{\mathbf{U}_{\rho(x),2}}\mathbf{B}_2^{*T} + \mathbf{c}_x^T\widehat{\mathbf{U}_{\rho(x),3}}\mathbf{B}_3^{*T} \\
&\quad= \mathbf{N}_x\begin{pmatrix}\gamma\mathbf{B}_3^{*T}\\ \mathbf{W}_{\mathsf{bot}}\end{pmatrix} + \mathbf{c}_x^T(k_3\widehat{\Delta})\mathbf{B}_3^{*T} + \mathbf{c}_x^T\widehat{\mathbf{U}_{\rho(x),1}}\mathbf{B}_1^{*T} + \mathbf{c}_x^T\widehat{\mathbf{U}_{\rho(x),2}}\mathbf{B}_2^{*T} + \mathbf{c}_x^T\widehat{\mathbf{U}'_{\rho(x),3}}\mathbf{B}_3^{*T} \\
&\quad= \mathbf{N}_x\begin{pmatrix}\gamma\mathbf{B}_3^{*T}\\ \mathbf{W}_{\mathsf{bot}}\end{pmatrix} + n_xk_3\tau\mathbf{B}_3^{*T} + \mathbf{c}_x^T\widehat{\mathbf{U}_{\rho(x),1}}\mathbf{B}_1^{*T} + \mathbf{c}_x^T\widehat{\mathbf{U}_{\rho(x),2}}\mathbf{B}_2^{*T} + \mathbf{c}_x^T\widehat{\mathbf{U}'_{\rho(x),3}}\mathbf{B}_3^{*T} \\
&\quad= \mathbf{N}_x\begin{pmatrix}\gamma\mathbf{B}_3^{*T}\\ \mathbf{W}_{\mathsf{bot}}\end{pmatrix} + \mathbf{N}_x\begin{pmatrix}k_3\tau\mathbf{B}_3^{*T}\\ \mathbf{0}\end{pmatrix} + \mathbf{c}_x^T\widehat{\mathbf{U}_{\rho(x),1}}\mathbf{B}_1^{*T} + \mathbf{c}_x^T\widehat{\mathbf{U}_{\rho(x),2}}\mathbf{B}_2^{*T} + \mathbf{c}_x^T\widehat{\mathbf{U}'_{\rho(x),3}}\mathbf{B}_3^{*T} \\
&\quad= \mathbf{N}_x\begin{pmatrix}(\gamma+k_3\tau)\mathbf{B}_3^{*T}\\ \mathbf{W}_{\mathsf{bot}}\end{pmatrix} + \mathbf{c}_x^T\widehat{\mathbf{U}'_{\rho(x)}} \\
&\quad= \mathbf{N}_x\begin{pmatrix}\gamma'\mathbf{B}_3^{*T}\\ \mathbf{W}_{\mathsf{bot}}\end{pmatrix} + \mathbf{c}_x^T\widehat{\mathbf{U}'_{\rho(x)}}
\end{aligned}
$$

where we can write $\gamma' = \gamma + k_3\tau$. Similarly, we can write

$$\mathbf{M}_x\mathbf{W} + \mathbf{c}_x^T\mathbf{U}_{\rho(x)} = \mathbf{M}_x\begin{pmatrix}\gamma'\mathbf{B}_3^{*T}\\ \mathbf{W}_{\mathsf{bot}}\end{pmatrix} + \mathbf{c}_x^T\mathbf{U}'_{\rho(x)}.$$

Therefore to complete this part of the proof, it suffices to argue that γ and γ' are identically distributed. This holds true because γ, k_3, τ are uniform random, that is, $\gamma, k_3, \tau \xleftarrow{\$} \mathbb{Z}_q$.

In conclusion, substituting $(\mathbf{V}_{\rho(x),3}, \mathbf{U}_{\rho(x),3}, \widehat{\mathbf{V}_{\rho(x),3}}, \widehat{\mathbf{U}_{\rho(x),3}}, \gamma)$ with $(\mathbf{V}'_{\rho(x),3}, \mathbf{U}'_{\rho(x),3}, \widehat{\mathbf{V}'_{\rho(x),3}}, \widehat{\mathbf{U}'_{\rho(x),3}}, \gamma')$ (as described above) for all rows $x \in [n]$ of matrices $\mathbf{M}, \mathbf{N}$ allows us to move from Hyb_4 to $\mathsf{Hyb}_{3,q}$.

Therefore, it follows that $\mathsf{Hyb}_{3,q}$ and Hyb_4 are statistically indistinguishable. This completes the proof of Claim D.14. □

Hybrid Hyb_5. This is same as Hyb_4 except that the ciphertext is changed to an encryption of a random value, that is, c_0 is changed from $c_0 := \mathsf{msg}_b \cdot [\![\mathbf{t}^T\mathbf{k}]\!]_T$ to $c_0 := \zeta \cdot [\![\mathbf{t}^T\mathbf{k}]\!]_T$, where $\zeta \xleftarrow{\$} \mathbb{G}_T$.

Claim D.15. *Hyb_4 and Hyb_5 are statistically indistinguishable.*

Proof. Similar to proof of Claim 4.12 □

Claim D.16. *In Hyb_5, adversary $\mathcal{A}$'s winning advantage is 0.*

Proof. This holds because Hyb_5 contains no information about challenge messages msg_0 and msg_1. □

Thus, from Claims D.5 to D.16 and hybrid argument, it follows that Theorem 6.1 holds.

$\mathsf{GlobalSetup}(1^\lambda)$:

1 : $\mathsf{igp} \leftarrow \mathsf{inner.GlobalSetup}(1^\lambda)$
2 : $\mathsf{ogp} \leftarrow \mathsf{outer.GlobalSetup}(1^\lambda)$
3 : **ret** $\mathsf{gp} := (\mathsf{igp}, \mathsf{ogp})$

$\mathsf{AuthSetup}(\mathsf{gp}, i)$:

1 : $(\mathsf{imsk}_i, \mathsf{ipk}_i) \leftarrow \mathsf{inner.AuthSetup}(\mathsf{igp}, i)$
2 : $(\mathsf{omsk}_i, \mathsf{opk}_i) \leftarrow \mathsf{outer.AuthSetup}(\mathsf{ogp}, i)$
3 : **ret** $\mathsf{msk}_i := (\mathsf{imsk}_i, \mathsf{omsk}_i), \mathsf{pk}_i := (\mathsf{ipk}_i, \mathsf{opk}_i)$

$\mathsf{KGen}(\mathsf{gp}, \mathsf{msk}_i, \mathsf{GID}, z_i)$:

1 : $\mathsf{isk}_{i,\mathsf{GID},z_i} \leftarrow \mathsf{inner.KGen}(\mathsf{igp}, \mathsf{imsk}_i, \mathsf{GID}, z_i)$
2 : $\mathsf{osk}_{i,\mathsf{GID}} \leftarrow \mathsf{outer.KGen}(\mathsf{ogp}, \mathsf{omsk}_i, \mathsf{GID})$
3 : **ret** $\mathsf{sk}_{i,\mathsf{GID},z_i} := (\mathsf{isk}_{i,\mathsf{GID},z_i}, \mathsf{osk}_{i,\mathsf{GID}})$

$\mathsf{Enc}(\mathsf{gp}, \mathsf{msg} \in \mathbb{G}_T, (\mathbf{M}, \mathbf{N}, \rho), \{\mathsf{pk}_{\rho(i)}\}_{i\in[n]})$:

1 : $\mathsf{ict} \leftarrow \mathsf{inner.Enc}(\mathsf{igp}, \mathsf{msg}, (\mathbf{M}, \mathbf{N}, \rho), \{\mathsf{ipk}_{\rho(i)}\}_{i\in[n]})$
2 : Let $U = \{\rho(i) : i \in [n]\}$ be the set of authorities appearing in the ASP policy
3 : Let $P_U =$ "conjunction of all authorities in set U" be a policy
4 : $\mathsf{oct} \leftarrow \mathsf{outer.Enc}(\mathsf{ogp}, \mathsf{ict}, P_U, \{\mathsf{opk}_i\}_{i\in U})$
5 : **ret** $\mathsf{ct} := \mathsf{oct}$

$\mathsf{Dec}(\mathsf{gp}, (\mathbf{M}, \mathbf{N}, \rho), \mathsf{ct}, \mathsf{GID}, \{\mathsf{sk}_{\rho(i),\mathsf{GID},z_{\rho(i)}}\}_{i\in[n]})$:

1 : Let $U = \{\rho(i) : i \in [n]\}$
2 : Let $P_U =$ "conjunction of all authorities in set U" be a policy
3 : $\mathsf{omsg} \leftarrow \mathsf{outer.Dec}(\mathsf{ogp}, P_U, \mathsf{ct}, \mathsf{GID}, \{\mathsf{osk}_{\rho(i),\mathsf{GID},z_{\rho(i)}}\}_{\rho(i)\in U})$
4 : $\mathsf{imsg} \leftarrow \mathsf{inner.Dec}(\mathsf{igp}, (\mathbf{M}, \mathbf{N}, \rho), \mathsf{omsg}, \mathsf{GID}, \{\mathsf{isk}_{\rho(i),\mathsf{GID},z_{\rho(i)}}\})$
5 : **ret** imsg

Figure 5: Compiler Construction: MA-ABE for ASP

E Generic Compiler for MA-ABE for ASP: boosting security: Appendix

In this section, we provide the formal construction of the MA-ABE scheme presented in Section 7. We also provide a proof sketch of its security.

Theorem E.1. *Suppose we have the following two MA-ABE schemes:*

- inner *denoting an MA-ABE for ASP that is fully adaptively secure with Type 1 restriction (Definition D.2),*
- outer *denoting an MA-ABE for conjunctions that is fully adaptively secure (Definition A.6).*

Then, MA-ABE scheme for ASP presented in Figure 5 is fully adaptive secure with Type 2 restriction (Definition D.3).

Proof Sketch. Security proof roughly works as follows. Consider hybrids H_0^b, H_1^b for $b \in \{0, 1\}$ as follows. H_0^b is real-world encrypting msg_b. H_1^b is same as H_0^b except that the game samples uniform random $\beta \in \{0, 1\}$ at the beginning, where $\beta = 1$ indicates a guess that event A will occur and $\beta = 0$ indicates a guess that event A will not occur, where event A is: the adversary will satisfy the condition "NO corrupt authority appears in challenge policy" at the end of the game. The rest of the game proceeds same as H_0^b except that the game aborts if the guess turns out to be wrong, that

is, one of two things occur: "$\beta = 1$ and not event A" or "$\beta = 0$ and event A". Next, we make two claims.

Claim E.2. *If there exists a p.p.t. adversary with distinguishing advantage ϵ in distinguishing between its view in H_0^0 and H_0^1, then, there exists a p.p.t. adversary with distinguishing advantage $\epsilon/2$ in distinguishing between its view in H_1^0 and H_1^1.*

Claim E.3. *If the inner scheme MA-ABE for ASP is weak adaptive secure and if the outer scheme MA-ABE for conjunction is fully adaptive secure, then, for any p.p.t. adversary, the distinguishing advantage in distinguishing between its view in H_1^0 and H_1^1 is negligible.*

From Claims E.2 and E.3, Theorem E.1 follows. All that remains to prove are Claims E.2 and E.3.

Claim E.2 follows from the fact that $\beta \in \{0, 1\}$ is sampled uniformly at random in H_1^b for $b \in \{0, 1\}$.

To see Claim E.3, suppose there exists an adversary that distinguishes hybrids H_1^0 and H_1^1 with non-negligible probability, then, we describe a reduction algorithm R that breaks either the weak adaptive security of MA-ABE for ASP or breaks fully adaptive security of MA-ABE for conjunction. The reduction R works as follows.

At the beginning, R samples uniform random $\beta \in \{0, 1\}$.

If $\beta = 1$, reduction R simulates MA-ABE for conjunction on its own and talks to challenger C_{ASP} for the weak adaptive security game of MA-ABE for ASP. $\mathsf{GlobalSetup}$ and oracles $\mathsf{AuthSetup}$, KGen, $\mathsf{Corrupt}$ are handled in the natural way. For challenge query $(m_0, m_1, (\mathbf{M}, \mathbf{N}, \rho))$, it forwards the query to the challenger C_{ASP} and obtains ciphertext ict encrypting message m_b. Then it re-encrypts ict with respect to policy "conjunction of all authorities in set U" to obtain ciphertext oct, where U is the set of authorities appearing in policy $(\mathbf{M}, \mathbf{N}, \rho)$. It returns oct to the adversary. Eventually the adversary outputs a guess bit b'. At this point if it is the case that adversary *did not* corrupt any authority appearing in the challenge ciphertext policy, then it forwards b' to the challenger C_{ASP}. Else it sends a random bit $\widetilde{b}$ to the challenger.

If $\beta = 0$, reduction R simulates inner MA-ABE for ASP with weak adaptive security on its own and talks to challenger $C_{conjunction}$ for the fully adaptive security game of outer MA-ABE for conjunction. $\mathsf{GlobalSetup}$ and oracles $\mathsf{AuthSetup}$, KGen, $\mathsf{Corrupt}$ are handled in the natural way. For challenge query $(m_0, m_1, (\mathbf{M}, \mathbf{N}, \rho))$, it locally computes ciphertexts ict_0 and ict_1 encrypting m_0 and m_1 respectively and sends challenge query $(\mathsf{ict}_0, \mathsf{ict}_1,$ "conjunction of all authorities in set U") to the challenger $C_{conjunction}$ and obtains ciphertext oct encrypting ict_b, where U is the set of authorities appearing in policy $(\mathbf{M}, \mathbf{N}, \rho)$. It returns oct to the adversary. Eventually the adversary outputs a guess bit b'. At this point if it is the case that adversary *did* corrupt any authority appearing in the challenge ciphertext policy, then it forwards b' to the challenger $C_{conjunction}$. Else it sends a random bit $\widetilde{b}$ to the challenger.

Observe that when $\beta = 0$, if the adversary is admissible and the guess β is correct, then, the reduction is admissible in its game against challenger C_{ASP}. Further, when $\beta = 1$, if the adversary is admissible and the guess β is correct, then, it must be the case that there exists an honest authority appearing in the challenge ciphertext policy who did not issue any secret key. Thus the conjunction policy is not satisfied and hence the reduction is admissible in its game against challenger $C_{conjunction}$. Hence, in both cases $\beta = 0$ and $\beta = 1$, the reduction perfectly simulates the views H_1^b to the adversary, where b is as chosen by the appropriate challenger (C_{ASP} or $C_{conjunction}$). Thus, it follows that if there exists an adversary that distinguishes hybrids H_1^0 and H_1^1 with non-negligible probability, then, the reduction algorithm R that breaks either the weak adaptive security of MA-ABE for ASP or breaks fully adaptive security of MA-ABE for conjunction. This completes the proof of Claim E.3 and hence the proof of Theorem E.1.

Table 2: Communication Efficiency: Comparison of fully adaptively secure decentralized MA-ABE in prime order groups. All schemes are based on k-MDDH assumption in prime-order asymmetric pairing groups, so we need $k \geq 1$. n denotes the number of rows in the policy matrix $\mathbf{M}$ (and also matrix $\mathbf{N}$ in case of ASP). In ciphertext

Scheme	$\lvert\mathsf{msk}_i\rvert$	$\lvert\mathsf{pk}_i\rvert$	$\lvert sk_{i,\mathsf{GID}}\rvert$	$\lvert\mathsf{ct}\rvert$	Many-use?
DKW23 [DKW23]	$18k^2\lvert\mathbb{Z}_q\rvert$	$6k^2\lvert\mathbb{G}_1\rvert$	$6k\lvert\mathbb{G}_2\rvert$	$12nk\lvert\mathbb{G}_1\rvert$	No
Ours Section 4	$(4k^2+6k+2)\lvert\mathbb{Z}_q\rvert$	$(4k^2+2k)\lvert\mathbb{G}_1\rvert$	$(k+1)\lvert\mathbb{G}_2\rvert$	$n(5k+3)\lvert\mathbb{G}_1\rvert$	No
Ours[4] Section 6	$(8k^2+12k+4)\lvert\mathbb{Z}_q\rvert$	$(8k^2+4k)\lvert\mathbb{G}_1\rvert$	$(k+1)\lvert\mathbb{G}_2\rvert$	$n(9k+5)\lvert\mathbb{G}_1\rvert$	No
CCG+23 [CCG+23]	$(12k^2+6k)\lvert\mathbb{Z}_q\rvert$	$6k^2\lvert\mathbb{G}_1\rvert$	$(4k+2)\lvert\mathbb{G}_2\rvert$	$n(10k+2)\lvert\mathbb{G}_1\rvert$	Yes
Ours Section 5	$(8k^2+4k)\lvert\mathbb{Z}_q\rvert$	$(4k^2+2k)\lvert\mathbb{G}_1\rvert$	$2k\lvert\mathbb{G}_2\rvert$	$n(6k+2)\lvert\mathbb{G}_1\rvert$	Yes

F Efficiency Analysis: Detailed

Notations for communication costs. We denote the size of the group $\mathbb{G}_1$ as $|\mathbb{G}_1|$, the size of the group $\mathbb{G}_2$ as $|\mathbb{G}_2|$, and the size of the field $\mathbb{Z}_q$ as $|\mathbb{Z}_q|$. We denote the size of the master secret key msk_i as $|\mathsf{msk}_i|$, the size of the public key pk_i as $|\mathsf{pk}_i|$, the size of the secret key $sk_{i,\mathsf{GID}}$ as $|sk_{i,\mathsf{GID}}|$, and the size of the ciphertext ct as $|\mathsf{ct}|$. We denote the number of rows in the policy matrix $\mathbf{M}$ as n, and the number of columns in the policy matrix $\mathbf{M}$ as ℓ. We denote the set of attributes with respect to which decryption is performed as S.

Notations for computation costs. For measuring computation costs, we note that there are three different types of operations, each with different costs: group operation, exponentiation, pairing. Typically group operation is faster than the other two by 3 orders of magnitude (See Table 4), so we can ignore it in the comparison. We denote the number of exponentiations as $\#\mathsf{exponentiations}$, and the number of pairings as $\#\mathsf{pairings}$. For $\#\mathsf{exponentiations}$, $(\mathsf{count})_i$ denotes the count for group G_i where $i \in \{1, 2, T\}$.

Asymptotic efficiency of our schemes is summarized in Table 2 and Table 3. We note that the communication and computation costs in Table 1 are obtained by substituting $k = 1$ in Tables 2 and 3.

Lastly, we present estimates of concrete computation costs in Table 4. To estimate concrete computation costs, we use the BLS12-381 elliptic curve as a representative for instantiating prime-order pairing groups. For the BLS12-381 elliptic curve, we use the microbenchmarks from [Tom22]. The numbers are for BLS12-381 curve implemented in Filecoin's blstrs [fil] Rust wrapper around the popular blst [sup] library. These microbenchmarks were run on a 10-core 2021 Apple M1 Max. The computation costs in Table 4 are obtained by multiplying the number of operations in Table 1b with the corresponding microbenchmark numbers in Table 5.

[4] Our MA-ABE for ASP satisfies weak adaptive security as defined in Definition D.2

Table 3: Computation Efficiency: Comparison of fully adaptively secure decentralized MA-ABE in prime order groups. All schemes are based on k-MDDH assumption in prime-order asymmetric pairing groups, so we need $k \geq 1$. n and ℓ denote the number of rows and columns in the policy matrix $\mathbf{M}$ (and also matrix $\mathbf{N}$ in case of ASP) respectively. S denotes the set of attributes with respect to which decryption is performed. For each algorithm, we specify a tuple of three values: #group ops, #exponentiations, #pairings. For #group ops and #exponentiations, $(\mathsf{count})_i$ denotes the count for group G_i where $i \in \{1, 2, T\}$.

Scheme	AuthSetup	KGen	Enc	Dec
DKW23 [DKW23]	$(18k^3)_1$, $(18k^3)_1$, 0	$(18k^2+3k)_2$, $(18k^2)_2$, 0	$(12nk^2+6nk\ell+3k^2+6nk)_1$, $(12nk^2+6nk\ell+3k^2+6(\ell-1)k)_1+(3k)_T$, $3k$	$(3k\|S\|)_2+((12k+4)\|S\|)_T$, $(\|S\|)_T$, $12k\|S\|$
Ours Section 4	$(4k^3+6k^2+2k)_1$, $(4k^3+6k^2+2k)_1$, 0	$(2k^2+3k+2)_2$, $(2k^2+3k+2)_2$, 0	$(n(5k^2+7k+2))_1+(1)_T$, $(n(5k^2+7k+2))_1+(1)_T$, 0	$((2k+1)\|S\|)_1+((3k+5)\|S\|+1)_T$, $((2k+1)\|S\|)_1+(\|S\|)_T$, $(3k+3)\|S\|$
Ours Section 6	$(8k^3+12k^2+4k)_1$, $(8k^3+12k^2+4k)_1$, 0	$(2k^2+3k+2)_2$, $(2k^2+3k+2)_2$, 0	$(n(9k^2+13k+4))_1+(1)_T$, $(n(9k^2+13k+4))_1+(1)_T$, 0	$((2k+1)\|S\|)_1+((7k+7)\|S\|+1)_T$, $((6k+3)\|S\|)_1+(\|S\|)_T$, $(3k+3)\|S\|$
CCG+23 [CCG+23]	$(12k^3+6k^2)_1$, $(12k^3+6k^2)_1$, 0	$(12k^2+9k)_2$, $(12k^2+6k)_2$, 0	$(n(10k^2+8k))_1$, $(n(10k^2+8k))_1+(3k)_T$, $3k$	$(3k\|S\|)_2+((10k+6)\|S\|)_T$, $(\|S\|)_T$, $(10k+2)\|S\|$
Ours Section 5	$(8k^3+4k^2)_1$, $(8k^3+4k^2)_1$, 0	$(4k^2+2k+1)_2$, $(4k^2+2k+1)_2$, 0	$(n(6k^2+6k+2))_1+(1)_T$, $(n(6k^2+6k+2))_1+(1)_T$, 0	$((2k+1)\|S\|)_1+((4k+4)\|S\|+1)_T$, $((2k+1)\|S\|)_1+(\|S\|)_T$, $(4k+2)\|S\|$

Table 4: Computation Efficiency over BLS12-381 elliptic curve: Comparison of fully adaptively secure decentralized MA-ABE in prime order groups. All schemes are instantiated from k-MDDH assumption in prime-order pairing groups, where $k = 1$. n and ℓ denote the number of rows and columns in the policy matrix $\mathbf{M}$ (and also matrix $\mathbf{N}$ in case of ASP) respectively. S denotes the set of attributes with respect to which decryption is performed.

Scheme	AuthSetup	KGen	Enc	Dec
DKW23 [DKW23]	1.296ms	2.448ms	$(0.432n\ell+0.864n+0.432\ell+2.742)$ms	$(6.332\|S\|)$ms
Ours Section 4	0.864ms	0.952ms	$(1.008n+0.5)$ms	$(3.632\|S\|)$ms
Ours Section 6	1.728ms	0.952ms	$(1.876n+0.5)$ms	$(4.064\|S\|)$ms
CCG+23 [CCG+23]	1.728ms	2.856ms	$(1.296n+4.458)$ms	$(6.332\|S\|)$ms
Ours Section 5	0.826ms	0.952ms	$(1.008n+0.5)$ms	$(3.632\|S\|)$ms

Table 5: Computation time of various operations: We recall the computation time of various operations on the BLS12-381 elliptic curve from [Tom22]. The numbers are BLS12-381 curve implemented in Filecoin's blstrs [fil] Rust wrapper around the popular blst [sup] library. These microbenchmarks were run on a 10-core 2021 Apple M1 Max.

Operation	$\mathbb{G}_1$	$\mathbb{G}_2$	$\mathbb{G}_T$
group op	565ns	1484ns	1617ns
exponentiation	72μs	136μs	500μs
pairing	486μs		