Decentralized Multi-Authority ABE for NC^1 from Computational-BDH

Pratish Datta[1], Ilan Komargodski[2], and Brent Waters[3]

November 3, 2026

Abstract

Decentralized multi-authority attribute-based encryption (MA-ABE) is a strengthening of standard ciphertext-policy attribute-based encryption so that there is no trusted central authority: any party can become an authority and there is no requirement for any global coordination other than the creation of an initial set of common reference parameters. Essentially, any party can act as an authority for some attribute by creating a public key of its own and issuing private keys to different users that reflect their attributes.

This paper presents the first MA-ABE proven secure under the standard *search* variant of bilinear Diffie-Hellman (CBDH) and in the random oracle model. Our scheme supports all access policies captured by NC^1 circuits.

All previous constructions were proven secure in the random oracle model and additionally were based on *decision* assumptions such as the DLIN assumption, non-standard q-type assumptions, or subspace decision assumptions over composite-order bilinear groups.

Erratum. Subsequent to our publication, Wang [Wan26] showed plaintext-recovery attacks on both our DBDH construction and its CBDH variant through collusion between corrupt authorities and users in the stated static security model. One such attack uses a three-attribute conjunction policy with a single corrupt authority colluding with two users, neither of whom is authorized even when given the corrupt authority's attribute. Our proof of security from Section 5.2 incorrectly applies Lemma 3.1, originally from [RW15], to conclude that replacing the LSSS matrix preserves the adversary's entire view. The lemma does preserve the distribution of the secret shares, but the matrix transformation need not preserve the distribution of the ciphertext components containing the individual products $M_{i,j}x_j$. Wang also shows how to modify our construction and recover our claimed results by replacing, for each index j, the products $M_{i,j}x_j$ across the rows with an independently sampled linear sharing of zero. Apart from this erratum and the annotation identifying the error in Section 5.2, we leave the rest of the paper as originally published.

[Wan26] Xingwang Wang. *Cryptanalysis and Repair of "Decentralized Multi-authority ABE for* NC^1 *from BDH"*. Cryptology ePrint Archive, Paper 2026/2642, 2026. https://eprint.iacr.org/2026/2642. See Sections 5, 6, and 7.

[1] NTT Research. Email: pratish.datta@ntt-research.com
[2] Hebrew University and NTT Research. Email: ilank@cs.huji.ac.il
[3] UT Austin and NTT Research. Email: bwaters@cs.utexas.edu

Table of Contents

Decentralized Multi-Authority ABE for NC^1 from DBDH 1
Pratish Datta, Ilan Komargodski, and Brent Waters
1 Introduction 1
1.1 Our Contribution 3
1.2 Additional Related Works 3
2 Techniques 4
3 Preliminaries 7
3.1 Bilinear Groups and Complexity Assumptions 8
3.2 Access Structures and Linear Secret Sharing Schemes 8
3.3 Decentralized MA-ABE for LSSS 10
4 Our MA-ABE Scheme from DBDH 12
5 Correctness and Security Analysis of Our MA-ABE Scheme 13
5.1 Correctness 13
5.2 Security Analysis 15

1 Introduction

Attribute-based encryption (ABE) is a generalization of traditional public-key encryption [DH76, RSA78, Gam85, Reg05] that offers fine-grained access control over encrypted data based on the credentials (or attributes) of the recipients. ABE comes in two avatars: *ciphertext-policy* and *key-policy*. In a ciphertext-policy ABE (CP-ABE), as the name suggests, ciphertexts are associated with access policies and keys are associated with attributes. In a key-policy ABE (KP-ABE), the roles of the attribute sets and the access policies are flipped, i.e., ciphertexts are associated with attributes and keys are associated with access policies. In both cases, decryption is possible *only when* the attributes satisfy the access policy. Moreover, it is required that given any ciphertext created with respect to an access policy, no group of colluding users none of whom individually possesses a secret key corresponding to an attribute set satisfying the access policy, should be able to decipher the encrypted message. This property is known as *collusion resistance*.

Since its inception by Goyal, Pandey, Sahai and Waters [SW05, GPSW06], ABE has become a fundamental cryptographic primitive with a long list of potential applications. Therefore, the task of designing ABE schemes has received tremendous attention by the cryptographic community resulting in a long sequence of works achieving various trade-offs between expressiveness, efficiency, security, and underlying assumptions [GPSW06, BSW07, OSW07, Wat09, LOS$^+$10, LW10, OT10, ALdP11, AFV11, LW11b, Wat11, LW12, OT12, Wat12, Boy13, GGH$^+$13, GVW13, Att14, BGG$^+$14, CW14, Wee14, CGW15, DDM15, KL15, Att16, AC16, BV16, CMM16, ABGW17, AC17a, AC17b, GKW17, CGKW18, Att19, AMY19, GWW19, KW19, Tsa19, AWY20, AY20, BV20, GW20, LL20a, LL20b, TA20, TKN20].

Multi-Authority ABE: There is one major limitation in a standard ABE scheme which was pointed out already in the original work of Sahai and Waters [SW05]. In an ABE scheme, each user must go to the single master authority and prove that he has a certain set of attributes in order to receive the secret keys corresponding to each of those attributes. This means we must have one trusted authority who monitors all attributes e.g. driver's licenses, voter registration, and college enrollment. In reality, however, there are different entities responsible for issuing and maintaining the different attributes e.g. the DMV is the controller of driver licenses and similarly the Board of Elections and the University office for the other two attributes, respectively. Therefore, we would want to be able to entrust each of the attributes to a different (and perhaps not entirely trusted) authority.

To address the above problem, Chase [Cha07] introduced the notion of *multi-authority* ABE (MA-ABE) schemes. In an MA-ABE, there are multiple authorities which control different attributes and each of them can issue secret keys to users possessing attributes under their control without any interaction with the other authorities in the system. Given a ciphertext generated with respect to some access policy, a user possessing a set of attributes satisfying the access policy can decrypt the ciphertext by pulling the individual secret keys it obtained from the various authorities controlling those attributes. The security requires collusion resistance against unauthorized users as described above with the important difference that now some of the attribute authorities may be corrupted and therefore may collude with the adversarial users.

Building MA-ABE schemes turned out to be somewhat challenging. After few initial attempts [Cha07, LCLS08, MKE08, CC09, MKE09] that had various limitations, Lewko and Waters [LW11a] were able to design the first truly decentralized MA-ABE scheme in which any party can become an authority and there is no requirement for any global coordination other than the creation of an initial trusted setup. In their scheme, a party can simply act as an authority by publishing a public key of its own and issuing private keys to different users that reflect their attributes. Different authorities need not even be aware of each other and they can join the system at any point of time. There is also no bound on the number of attribute authorities that can ever come into play during the lifetime of the system. Their scheme supports all access policies

computable by NC^1 circuits. For security, they rely on the random oracle model and additionally they work with composite-order bilinear groups whose order is the product of three primes and its security is derived under various instances of the general subgroup decision assumptions and another new computational assumption introduced by them. This implies that fairly large parameters will be needed in order to have a meaningful level of security. Consequently, it became desirable to design decentralized MA-ABE schemes in bilinear groups of prime order which provide drastically better performance compared to their composite-order counterparts [Len01, Gui13].

There exist pretty generic methods that translate composite-order-group-based systems to analogous prime-order-group-based system (e.g., Freeman [Fre10], Lewko [Lew12], and their many followups [OT10, OT12, KL15, Att16, AC16, CGKW18]) which could apply to the Lewko-Waters [LW11a] scheme. In fact, this pathway has been successfully accomplished by Okamoto and Takashima [OT20] resulting in a concrete prime-order analogue of the Lewko-Waters construction. This translation technique eventually results with schemes that rely on the decisional linear (DLIN) assumption [BBS04] and its generalizations. One downside of this translation methodology is that it essentially tries to simulate the subgroup structure in the composite-order setting by developing vector space structures and thus incurs additional overheads which hinders the potentially efficiency enhancements of the prime-order bilinear group setting. In an attempt to avoid such additional overheads, Rouselakis and Waters [RW15] presented a different prime-order-group-based direct construction which has additional efficiency improvements over the Lewko-Waters construction by allowing each authority to control an arbitrary number of attributes (as opposed to only a single attribute in [LW11a]) and by supporting the use of an attribute an arbitrary number of times inside an access policy (contrary to the single-use restriction in [LW11a]). However, for security they rely (in addition to a random oracle) on a newly introduced (complicated and rather non-standard) q-type assumption.

Motivation: In this work we are interested in basing MA-ABE for expressive access policies under more standard or "simpler" assumptions. In particular,

Can we base MA-ABE *for expressive access policies, e.g. for* NC^1*, under the same assumptions as standard "centralized"* ABE *schemes for such policies are based on?*

One prominent gap that remains with respect to this question is basing MA-ABE for expressive access policies, e.g. NC^1, on the decisional or computational bilinear Diffie-Hellman assumptions (DBDH or CBDH [BF01])[1], similarly to the centralized ABE [SW05, GPSW06]. All assumptions previously used to get MA-ABE are potentially stronger and much less standard: they are either (1) using novel q-type assumptions which in addition to being less standard are known to be non-trivially vulnerable to attacks [Che06, RW15], or (2) decisional assumptions with no natural "search" variant and where the target of the assumption is in the source group [LW11a, OT20].[2]

Specifically, in DLIN the target of the group is in the source group while in C/DBDH the target is in the target group.[3] This difference might seem syntactic, but experience has shown that techniques involved in target-based assumptions can often be translated to other settings, such as ones based on lattices. Further, DLIN-style assumptions have no search-version analog (as CBDH is to DBDH).

Additionally, q-type assumptions are vulnerable to non-trivial attacks that recover the secret involved in a q-type assumption in time inversely proportional to q [KOS01, Ver01, Jou04, Che06, SHI+12]. Hence, the parameters of any q-type-assumption-based cryptographic constructions

[1] The decision version of BDH asks to distinguish $e(g,g)^{abc}$ from a random (target) group element given random (g, g^a, g^b, g^c), while the search version asks to compute $e(g,g)^{abc}$ given (g, g^a, g^b, g^c).

[2] If $e\colon \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is a bilinear map, then we refer to elements in $\mathbb{G}$ as being in the source group or bilinear group.

[3] In DLIN it is assumed to be hard to distinguish between g^{a+b} from a random (source) group element given random elements (g, v, w, v^a, w^b) in the source group.

must scale with q. In particular, in the case of [RW15], the size of the q-type assumption scales with the complexity of the access policies associated with the ciphertext in the scheme which means that all parameters must grow with the complexity of the supported access policies.

Lastly, we mention that a recent work of Datta et al. [DKW21] gave a construction of MA-ABE based on Learning With Errors (LWE) that overcomes the aforementioned two problems (since LWE is a standard assumption and its decision version reduces to search). However, it supports only DNF access policies.

1.1 Our Contribution

We close the aforementioned gap by building the first decentralized MA-ABE scheme for NC^1 access policies relying only on the *simple static search* BDH assumption (and in the random oracle model, as all previous construction). Similarly to [LW11a, RW15, OT20], in our MA-ABE scheme, any party can become an authority at any point of time and there is no bound on the number of attribute authorities that can join the system or need for any global coordination other than the creation of an initial set of common reference parameters created during a trusted setup. Like [RW15], we only obtain static security where all of the ciphertexts, secret keys, and corruption queries must be issued by the adversary before the public key of any attribute authority is published.[4]

Theorem 1.1 (informal): *There exists a statically-secure decentralized* MA-ABE *scheme supporting all access policies captured by* NC^1 *circuits in prime-order bilinear groups in the random oracle model assuming the computational bilinear Diffie-Hellman* (CBDH) *assumption.*

1.2 Additional Related Works

While not relevant in the context of this paper, we would like to mention briefly about some recent works on MA-ABE [Kim19, WFL19, DKW21]. All these constructions are proven secure under the Learning with Errors (LWE) assumption. Out of the three constructions, the ones by Kim [Kim19] and Wang et al. [WFL19] assume a central authority which generates the public and secret keys for all the attribute authorities in the system. Thus all authorities that will ever exist in the system are forever fixed once setup is complete which runs counter to the truly decentralized spirit we consider in this paper. Additionally, both schemes guarantee security only against a bounded collusion of parties. In fact, the scheme of Kim [Kim19] is built in a new model, called the "OT model", which is incapable of handling even bounded collusion as noted in [DKW21]. In this sense, both the constructions [Kim19, WFL19] suffer from related limitations to the early MA-ABE constructions [Cha07, LCLS08, MKE08, CC09, MKE09]. Overcoming the limitations of [Kim19, WFL19], Datta, Komargodski, and Waters [DKW21] put forward the first truly decentralized MA-ABE scheme secure against arbitrary collusions even in the presence of malicious attribute authorities under the LWE assumption. The expressiveness of the three constructions is different, namely, the scheme of Datta, Komargodski, and Waters [DKW21] supports access policies captured by DNFs formulas while those of Wang et al. [WFL19] and Kim [Kim19] support access policies captured by NC^1 and arbitrary bounded depth circuits, respectively.

Paper Organization: In Section 2, we provide a high-level overview of our techniques. Definitions of linear secret sharing schemes, and the definitions of an MA-ABE are provided in Section 3. In Section 4, we present our MA-ABE scheme. The proofs of correctness and security are differed to Section 5. In Section 1.2, we mention some lattice-based recent MA-ABE constructions

[4] Note that currently, the only known technique to achieve adaptive security for ABE is Waters' "dual system encryption" methodology [Wat09, LW10] which crucially relies on the hidden subgroup or subspace structure.

and in Section 3.1 we recall the syntax of bilinear groups as well as our hardness assumptions (DBDH/CBDH).

2 Techniques

In this section, we will describe the challenges towards instantiating decentralized MA-ABE under the decisional BDH assumption and our main ideas to overcome those challenges. The construction based on the computational (search) variant of BDH (i.e., CBDH) follows by standard techniques involving a hardcore bit function for the CBDH problem.

Background: Before explaining how we obtain our construction, let us give some background on MA-ABE schemes. Our MA-ABE (like all other known MA-ABE schemes) is designed under the assumption that each user in the system has a unique global identifier GID coming from some universe of global identifiers $\mathcal{GID} \subset \{0,1\}^*$. We shall further assume (without loss of generality) that each authority controls just one attribute, and hence we can use the words "authority" and "attribute" interchangeably. We denote the authority universe by $\mathcal{AU}$.

First, let us recall the syntax of decentralized MA-ABE for NC^1 access policies, which is well known to be realizable by (monotone) linear secret sharing schemes (LSSS) [BL88, LW11a]. A decentralized MA-ABE scheme consists of 5 procedures GlobalSetup, AuthSetup, KeyGen, Enc, and Dec. The GlobalSetup procedure gets as input the security parameter (in unary encoding) and outputs global parameters. All of the other procedures depend on these global parameters (we may sometimes not mention them explicitly when they are clear from context). The AuthSetup procedure can be executed by any authority $u \in \mathcal{AU}$ to generate a corresponding public and secret key pair, PK_u and SK_u. At this point, an authority holding the secret key SK_u can generate a secret key $\mathsf{SK}_{\mathsf{GID},u}$ for a user with global identifier GID. At any point in time, using the public keys $\{\mathsf{PK}_u\}$ of some authorities, one can encrypt a message msg relative to some linear secret sharing policy $(\boldsymbol{M}, \rho)$, where $\boldsymbol{M}$ is the policy matrix and ρ is the function that assigns row indices in the matrix to attributes controlled by those authorities, to get a ciphertext CT. Finally, a user holding a set of secret keys $\{\mathsf{SK}_{\mathsf{GID},u}\}$ (relative to the same GID) can decrypt a given ciphertext CT if and only if the attributes corresponding to the secret it possesses "satisfy" the access structure with which the ciphertext was generated. If the MA-ABE scheme is built in the random oracle model as is the case in this paper and in all previous collusion resistant MA-ABE schemes, the existence of a public hash function H mapping the global identifiers in $\mathcal{GID}$ to some appropriate space is considered. This hash function H is generated by GlobalSetup and is modeled as a random oracle in the security proof.

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 attribute 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.

Challenges and Inspirations from Prior Works: The main challenge in any ABE scheme is to design it in some way that is collusion resistant, as described above. The standard technique to achieve this is to use the randomness of KeyGen to tie together the secret key components corresponding to the various attributes of a user to stop collusion attacks by mixing and combining the secret key components held by multiple users.

In the centralized setting, where there is a single authority holding the master secret key and generating keys for particular sets of attributes, this is usually done by tying together the different key components representing the different attributes of a user with the help of fresh randomness specific to that user. Such randomization would make the different key components of a user compatible with each other, but not with the parts of a key issued to another user. This seems to be problematic in the decentralized, multi-authority, ABE setting since there is no single master authority generating randomness. More precisely, a decentralized MA-ABE attempts to simultaneously allow anyone to generate authority keys (without even being aware of the existence of other authorities in the system) and at the same time maintain collusion resistance.

Previous decentralized MA-ABE schemes [LW11a, RW15] use the output of a public hash function H applied on the user's global identity (the GID) as the randomness tying together multiple key components issued by different authorities. This is what necessitates the use of the random oracle model, that is, assume the hash function H can actually output good randomness. However, this means that the randomness responsible for tying together the different key components must be publicly computable (even by the attacker). This public computability requirement clearly makes this strategy of tying together the different key components of a user more challenging compared to the centralized, single-authority setting.

This challenge has been overcome in prior decentralized MA-ABE schemes [LW11a, RW15] as follows. Those schemes consider a random oracle H that maps each global identifier GID to a (bilinear) group element. Their idea was then to structure the decryption mechanism at each node ν of the policy circuit associated with the ciphertext such that a user will recover a target group element of the form $e(g,g)^{\mathsf{sh}_\nu} \cdot e(g,\mathsf{H}(\mathsf{GID}))^{\mathsf{sh}'_\nu}$. This group element first contains a secret share sh_ν of a secret z in the exponent, and these shares can be combined to reconstruct the secret and recover the message. Each of these is "blinded" by a share sh'_ν which is a share of 0 in the exponent with base $e(g,\mathsf{H}(\mathsf{GID}))$. This structure allows for the decryption algorithm to both reconstruct the main secret z and to unblind it in parallel. If a user with a particular identifier GID satisfies the access tree, it can reconstruct z in the exponent by raising the group elements to the proper reconstruction coefficients. At the same time, this operation will reconstruct the share of 0 and thus the $e(g,\mathsf{H}(\mathsf{GID}))$ terms will cancel out. Intuitively, if two users with different global identifiers $\mathsf{GID}, \mathsf{GID}'$ attempt to collude, the cancellation will not work since the sh'_ν shares will have different bases.

While arguing security, the absence of coordination among multiple attribute authorities and the existence of malicious ones also makes the reduction more challenging compared to a single-authority ABE scheme. We borrow ideas from the proof techniques of Rouselakis and Waters [RW15]. Recall that they achieved (static) security under a q-type assumption by extending the "program and cancel" technique in an application of the "partitioning" methodology [SW05, GPSW06, Wat11]. Roughly speaking, in the partitioning technique the simulator of the reduction sets up the public parameters of the systems in such a way that the powerset of the attribute universe is partitioned into two disjoint sets: One for which the simulator can create the set secret keys and answer the attacker's queries, and one for which this is not possible, where the challenge query should belong. Since Rouselakis and Waters [RW15] considered the static security model, the simulator knows in advance the required challenge set and therefore the suitable partition. However, since in a decentralized MA-ABE the public keys of corrupt authorities are generated by the adversary and thus the simulator cannot program them directly. Therefore, it is necessary to somehow make the components of the challenge ciphertext corresponding to the corrupted authorities independent of the secret used to generate the challenge ciphertext.

Rouselakis and Waters [RW15] resolved this challenge by introducing an information-theoretic transformation that converts any linear secret sharing policy $(\boldsymbol{M}, \rho)$ and any subset of rows in $\boldsymbol{M}$ which corresponds to an unauthorized set into another linear secret sharing policy $(\boldsymbol{M}', \rho)$ where some of the columns of the unauthorized rows are zeroed-out. This transformation allows

the simulator of the security reduction to isolate the corrupted rows of the challenge access policy and essentially ignore it for the simulation.

Remark 2.1 (Static vs. Selective Security): While the partitioning technique was successful in establishing selective security, that is, security against static ciphertext and adaptive key queries in case of single-authority ABE schemes, it can only support static key queries for MA-ABE. This is because in case of single-authority ABE, the secret key queries are naturally atomic, that is when a secret key query for a user is requested to the master authority, all the attributes possessed by that user are submitted at the same time. In contrast, in an MA-ABE scheme, when a user requests a secret key for some attribute to the authority controlling that attribute, it is not required to disclose the other attributes it possesses to the authority. Thus, adaptive key queries in the context of MA-ABE means that the adversary can adaptively request the different secret key components with respect to the same GID at different points of time without disclosing the other attributes for which it will request secret keys with respect to that GID later. This makes the programming of the secret keys seemingly difficult for the simulator.

Our Approach for Basing Security Under DBDH: In designing our MA-ABE scheme under DBDH, we attempt to implement an analogous structure of the decryption algorithm as that of [LW11a, RW15], as described above. That is, we want to blind each share of the secret masking the message in the ciphertext with a share of 0 in the exponent of a base that involves the hashed output of a particular GID so that decryption succeeds only when all secret keys used to decrypt a ciphertext correspond to the same GID. However, the primary challenge with realizing the above idea provably under DBDH is that we need a way for a reduction to embed the challenge access policy into the authority public keys in order to implement the partitioning strategy. Since the DBDH assumption gives the reduction much less components to do this compared to a q-type assumption such as one used in [RW15], there is no obvious path for reducing the construction of [RW15] to DBDH. We surmount this obstacle by expanding our ciphertext and authority public key spaces. Our construction is parameterized by an integer $s_{\max}$ that specifies the maximum number of columns in a linear secret sharing matrix, or alternatively, a bound on the number of AND gates in the corresponding NC^1 access policy [BL88,LW11a]. The authority public keys and ciphertext all grow linearly in $s_{\max}$. This expansion of the authority public key and ciphertext spaces in turn requires us to expand the output space of the oracle H. More precisely, we consider $s_{\max}$ random group elements for each GID which are defined as the output of H on $(\mathsf{GID}\|j)$ for all $j \in [s_{\max}]$. The lack of components in a DBDH problem instance, as opposed to a q-type assumption, also requires us to restrict the row-labeling function ρ of the linear secret sharing policies $(\boldsymbol{M}, \rho)$ to be injective which means that each attribute can appear within an access policy at most once. However, our scheme can be alleviated to one which allows an attribute to appear within an access policy a bounded number of times using the simple encoding technique devised by [Wat11, LW11a]. Our scheme is described as follows.

Our scheme is based on a bilinear group $\mathsf{G} = (q, \mathbb{G}, \mathbb{G}_T, g, e)$ of prime order q with a generator g where we shall assume that the DBDH assumption holds. The description of the bilinear group G as well as the description of H are part of the global setup.

In our scheme, whenever an authority $u \in \mathcal{AU}$ wants to create a secret and public key pair, it chooses $\alpha_u, y_{u,2}, \ldots, y_{u,s_{\max}} \leftarrow \mathbb{Z}_q$ and outputs the public key PK_u and the master secret key MSK_u as

$$\mathsf{PK}_u = (e(g,g)^{\alpha_u}, g^{y_{u,2}}, \ldots, g^{y_{u,s_{\max}}}) \qquad \mathsf{MSK}_u = (\alpha_u, y_{u,2}, \ldots, y_{u,s_{\max}}).$$

Then, to generate a certain secret key $\mathsf{SK}_{\mathsf{GID},u}$ for an authority $u \in \mathcal{AU}$ and a user's global identifier GID, the authority computes

$$\mathsf{SK}_{\mathsf{GID},u} = g^{\alpha_u} \prod_{j=2}^{s_{\max}} \mathsf{H}(\mathsf{GID} \parallel j)^{y_{u,j}}.$$

To encrypt a message msg relative to an access policy $(\boldsymbol{M}, \rho)$, where $\boldsymbol{M}$ is a $\ell \times s_{\max}$ matrix whose ith row is denoted $\boldsymbol{M}_i$, we sample

1. $r_1, \ldots, r_\ell \leftarrow \mathbb{Z}_q$,
2. $\boldsymbol{v} = (z, v_2, \ldots, v_{s_{\max}}) \leftarrow \mathbb{Z}_q^{s_{\max}}$ (for secret sharing the random mask z), and
3. $\boldsymbol{x} = (x_2, \ldots, x_{s_{\max}}) \leftarrow \mathbb{Z}_q^{s_{\max}-1}$ (for secret sharing 0),

and output $\mathsf{CT} = ((\boldsymbol{M}, \rho), C_0, \{C_{1,i}\}_{i\in[\ell]}, \{C_{2,i}\}_{i\in[\ell]}, \{C_{3,i,j}\}_{i\in[\ell], j\in\{2,\ldots,s_{\max}\}})$, where

$$
\begin{aligned}
&C_0 = e(g,g)^z \mathsf{msg} \\
&\forall i \in [\ell]\colon\ C_{1,i} = e(g,g)^{\boldsymbol{M}_i \cdot \boldsymbol{v}} e(g,g)^{\alpha_{\rho(i)} r_i}, \quad C_{2,i} = g^{r_i} \\
&\forall i \in [\ell], j \in \{2, \ldots, s_{\max}\}\colon\ C_{3,i,j} = g^{y_{\rho(i),j} r_i} g^{M_{i,j} x_j}.
\end{aligned}
$$

Finally, for decryption we verify that the given set of keys $\{\mathsf{SK}_{\mathsf{GID},u}\}$ is authorized to decrypt the given ciphertext CT. If not, we abort. Denote by I the row indices in $\boldsymbol{M}$ which are available given the available secret keys. Then, we find scalars $\{w_i \in \mathbb{Z}_q\}_{i\in[I]}$ such that $\sum_{i\in[I]} w_i \boldsymbol{M}_i = (1, 0, \ldots, 0)$ (which should exist since the set is authorized and we use linear secret sharing policy). Then, we compute and output

$$
C_0 \Bigg/ \prod_{i\in I} \left[\frac{C_{1,i} \cdot \prod_{j=2}^{s_{\max}} e(\mathsf{H}(\mathsf{GID} \parallel j), C_{3,i,j})}{e(\mathsf{SK}_{\mathsf{GID},\rho(i)}, C_{2,i})} \right]^{w_i}.
$$

Looking at the decryption equation above, observe that for each row $i \in I$ we recover

$$
e(g,g)^{\boldsymbol{M}_i \cdot \boldsymbol{v}} \overbrace{\prod_{j=2}^{s_{\max}} e(g, \mathsf{H}(\mathsf{GID} \| j))^{M_{i,j} x_j}}^{\text{blinding factor}}.
$$

If all these terms correspond to the same GID, then by exponentiating with the appropriate reconstruction coefficients $\{w_i\}_{i\in I}$, we can recover the random mask z in the exponent of $e(g,g)$ and also do away with the blinding factors $\prod_{j=2}^{s_{\max}} e(g, \mathsf{H}(\mathsf{GID} \| j))^{M_{i,j} x_j}$.

For proving security of the above construction, we first apply the same information-theoretic transformation as [RW15] to reach a state where we can essentially ignore the rows of the challenge access matrix controlled by the corrupt authorities. After that, we carefully embed the DBDH instance into the public keys of uncorrupt authorities and components of the challenge ciphertext corresponding to those uncorrupt authorities. We partition the powerset of the set of uncorrupt authorities appearing in the challenge access policy by carefully embedding the challenge DBDH instance in such a way that we are able to simulate secret keys for any GID and subsets of those authorities which are unauthorized when combined with the rows controlled by the corrupt authorities. For the details, please refer to the full security proof in Section 5.2.

3 Preliminaries

Throughout this paper we will denote the underlying security parameter by λ. A function $\mathsf{negl}\colon \mathbb{N} \to \mathbb{R}$ is *negligible* if it is asymptotically smaller than any inverse-polynomial function, namely, for every constant $c > 0$ there exists an integer N_c such that $\mathsf{negl}(\lambda) \leq \lambda^{-c}$ for all $\lambda > N_c$. We let $[n] = \{1, \ldots, n\}$.

Let PPT stand for probabilistic polynomial-time. For a distribution $\mathcal{X}$, we write $x \leftarrow \mathcal{X}$ to denote that x is sampled at random according to distribution $\mathcal{X}$. For a set X, we write $x \leftarrow X$ to denote that x is sampled according to the uniform distribution over the elements of X. Also for

any set X, we denote by $|X|$ and 2^X the cardinality and the power set of the set X respectively. We use bold lower case letters, such as $\boldsymbol{v}$, to denote vectors and upper-case, such as $\boldsymbol{M}$, for matrices. We assume all vectors, by default, are row vectors. The ith row of a matrix is denoted $\boldsymbol{M}_i$ and analogously for a set of row indices I, we denote $\boldsymbol{M}_I$ for the submatrix of $\boldsymbol{M}$ that consists of the rows $\boldsymbol{M}_i$ for all $i \in I$.

For an integer $q \geq 2$, we let $\mathbb{Z}_q$ denote the ring of integers modulo q. We represent $\mathbb{Z}_q$ as integers in the range $(-q/2, q/2]$. The set of matrices of size $m \times n$ with elements in $\mathbb{Z}_q$ is denoted by $\mathbb{Z}_q^{m \times n}$. Special subsets are the set of row vectors of length $n : \mathbb{Z}_q^{1 \times n}$, and column vectors of length $n : \mathbb{Z}_q^{n \times 1}$. We denote by $\boldsymbol{v} \cdot \boldsymbol{w}$ the inner product of vector $\boldsymbol{v}$ and $\boldsymbol{w}$, where each vector can either be a row or a column vector. The operation $(\cdot)^\top$ denotes the transpose of vectors/matrices.

3.1 Bilinear Groups and Complexity Assumptions

Our MA-ABE construction works with instantiations of bilinear groups of prime order. Abstractly, let $\mathbb{G}$ and $\mathbb{G}_T$ be two multiplicative cyclic groups of prime order $q = q(\lambda)$, where the group operation is efficiently computable in the security parameter λ. Let g be a generator of $\mathbb{G}$ and $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be an efficiently computable pairing function that satisfies the following properties:

- *Bilinearity*: for all $u, v \in \mathbb{G}$ and $a, b \in \mathbb{Z}_q$ it is true that $e(u^a, v^b) = e(u, v)^{ab}$.
- *Non-degeneracy*: $e(g, g) \neq 1_{\mathbb{G}_T}$, where $1_{\mathbb{G}_T}$ is the identity element of the group $\mathbb{G}_T$.

Let $\mathcal{G}$ be an algorithm that takes as input 1^λ, the unary encoding of the security parameter λ, and outputs the description of a bilinear group $\mathsf{G} = (q, \mathbb{G}, \mathbb{G}_T, g, e)$.

Our security proof of our proposed MA-ABE scheme is based on the decisional bilinear Diffie-Hellman (DBDH) assumption. Moreover, our scheme can be readily translated into one with security under the computational bilinear Diffie-Hellman (CBDH) assumption. These assumptions were introduced by Boneh and Franklin [BF01]. The CBDH assumption is weaker compared to DBDH in the sense that DBDH implies CBDH, but not vice versa. These assumptions are defined below.

Assumption 1 (Decisional/Computational Bilinear Diffie-Hellman (DBDH/CBDH)): For a security parameter $\lambda \in \mathbb{N}$, let $\mathsf{G} = (q, \mathbb{G}, \mathbb{G}_T, g, e) \leftarrow \mathcal{G}(1^\lambda)$ be a bilinear group. The DBDH assumption states that for any PPT adversary $\mathcal{A}$, there exists a negligible function negl such that for any security parameter $\lambda \in \mathbb{N}$

$$\begin{aligned}\mathsf{Adv}_{\mathcal{A}}^{\mathsf{DBDH}}(\lambda) = &\Big| \Pr\Big[1 \leftarrow \mathcal{A}(1^\lambda, \mathsf{G}, g^a, g^b, g^c, \tau) \mid \mathsf{G} \leftarrow \mathcal{G}(1^\lambda);\, a, b, c \leftarrow \mathbb{Z}_q;\, \tau = e(g,g)^{abc}\Big] \\ &- \Pr\Big[1 \leftarrow \mathcal{A}(1^\lambda, \mathsf{G}, g^a, g^b, g^c, \tau) \mid \mathsf{G} \leftarrow \mathcal{G}(1^\lambda);\, a, b, c \leftarrow \mathbb{Z}_q;\, \tau \leftarrow \mathbb{G}_T\Big]\Big| \\ \leq &\mathsf{negl}(\lambda).\end{aligned}$$

On the other hand, the CBDH assumption states that for any PPT adversary $\mathcal{A}$, there exists a negligible function negl such that for any security parameter $\lambda \in \mathbb{N}$

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{CBDH}}(\lambda) = \Pr\Big[e(g,g)^{abc} \leftarrow \mathcal{A}(1^\lambda, \mathsf{G}, g^a, g^b, g^c) \mid \mathsf{G} \leftarrow \mathcal{G}(1^\lambda);\, a, b, c \leftarrow \mathbb{Z}_q\Big] \leq \mathsf{negl}(\lambda)$$

3.2 Access Structures and Linear Secret Sharing Schemes

In this subsection, we present the formal definitions of access structures and linear secret-sharing schemes.

Definition 3.1 (Access Structures): 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 $C \in \mathbb{A}$.

Definition 3.2 (Linear Secret Sharing Schemes (LSSS)): Let $q = q(\lambda)$ be a prime and $\mathbb{U}$ the attribute universe. A secret sharing scheme Π with domain of secrets $\mathbb{Z}_q$ for a monotone access structure $\mathbb{A}$ over $\mathbb{U}$, a.k.a. a monotone secret sharing scheme, is a randomized algorithm that on input a secret $z \in \mathbb{Z}_q$ outputs $|\mathbb{U}|$ shares $\mathsf{sh}_1, \ldots, \mathsf{sh}_{|\mathbb{U}|}$ such that for any set $S \in \mathbb{A}$ the shares $\{\mathsf{sh}_i\}_{i \in S}$ determine z and other sets of shares are independent of z (as random variables). A secret-sharing scheme Π realizing monotone access structures on $\mathbb{U}$ is linear over $\mathbb{Z}_q$ if

1. The shares of a secret $z \in \mathbb{Z}_q$ for each attribute in $\mathbb{U}$ form a vector over $\mathbb{Z}_q$.
2. For each monotone access structure $\mathbb{A}$ on $\mathbb{U}$, there exists a matrix $\boldsymbol{M} \in \mathbb{Z}_q^{\ell \times s}$, called the share-generating matrix, and a function ρ: $[\ell] \to \mathbb{U}$, that labels the rows of $\boldsymbol{M}$ with attributes from $\mathbb{U}$ which satisfy the following: During the generation of the shares, we consider the vector $\boldsymbol{v} = (z, r_2, ..., r_s)$, where $r_2, \ldots, r_s \leftarrow \mathbb{Z}_q$. Then the vector of ℓ shares of the secret z according to Π is given by $\boldsymbol{\mu} = \boldsymbol{M}\boldsymbol{v}^\top \in \mathbb{Z}_q^{\ell \times 1}$, where for all $j \in [\ell]$ the share μ_j "belongs" to the attribute $\rho(j)$. We will be referring to the pair $(\boldsymbol{M}, \rho)$ as the LSSS policy of the access structure $\mathbb{A}$.

The correctness and security of a monotone LSSS are formalized in the following: Let S (resp. S') denote an authorized (resp. unauthorized) set of attributes according to some monotone access structure $\mathbb{A}$ and let I (resp. I') be the set of rows of the share generating matrix $\boldsymbol{M}$ of the LSSS policy pair $(\boldsymbol{M}, \rho)$ associated with $\mathbb{A}$ whose labels are in S (resp. S'). For correctness, there exist constants $\{w_i\}_{i \in I}$ in $\mathbb{Z}_q$ such that for any valid shares $\left\{\boldsymbol{\mu}_i = (\boldsymbol{M}\boldsymbol{v}^\top)_i\right\}_{i \in I}$ of a secret $z \in \mathbb{Z}_q$ according to Π, it is true that $\sum_{i \in I} w_i \boldsymbol{\mu}_i = z$ (equivalently, $\sum_{i \in I} w_i \boldsymbol{M}_i = (1, \overbrace{0, \ldots, 0}^{s-1})$, where $\boldsymbol{M}_i$ is the ith row of $\boldsymbol{M}$). For soundness, there are no such w_i's, as above. Additionally, there exists a vector $\boldsymbol{d} \in \mathbb{Z}_q^{1 \times s}$, such that its first component $d_1 = 1$ and $\boldsymbol{M}_i \cdot \boldsymbol{d} = 0$ for all $i \in I'$.

Remark 3.1 (NC^1 and Monotone LSSS): Consider an access structure $\mathbb{A}$ described by an NC^1 circuit. There is a folklore transformation that can convert this circuit by a Boolean formula of logarithmic depth that consists of (fan-in 2) AND, OR, and (fan-in 1) NOT gates. We can further push the NOT gates to the leaves using De Morgan laws, and assume that internal nodes only constitute of OR and AND gates and leaves are labeled either by attributes or their negations. In other words, we can represent any NC^1 policy over a set of attributes into one described by a monotone Boolean formula of logarithmic depth over the same attributes and their negations. Lewko and Waters [LW11a] presented a monotone LSSS for access structures described by monotone Boolean formulas. This implies that any NC^1 access policy can be captured by a monotone LSSS. Therefore, in this paper, we will only focus on designing an MA-ABE scheme for monotone LSSS

We will use the following information theoretic property of LSSS access policies in the security proof of our MA-ABE scheme. This lemma first appeared in [RW15, Lemma 1]. Recently, Datta, Komargodski, and Waters [DKW20] observed a gap in the proof of [RW15] and presented a corrected proof; for details see [DKW20, Section 4.3]. This lemma allows the simulator of our reduction to isolate an unauthorized set of rows of the challenge LSSS matrix submitted by the adversary and essentially ignore it throughout the security reduction. Like [RW15, DKW20], in our case as well, the rows of the challenge LSSS matrix corresponding to the corrupt authorities will constitute the unauthorized set in the application of the lemma.

Lemma 3.1: *Let $(\boldsymbol{M}, \rho)$ be an LSSS access policy, where $\boldsymbol{M} \in \mathbb{Z}_q^{\ell \times s}$. Let $\mathcal{C} \subset [\ell]$ be a non-authorized subset of row indices of $\boldsymbol{M}$. Let $c \in \mathbb{N}$ be the dimension of the subspace spanned by*

the rows of $\boldsymbol{M}$ *corresponding to indices in* $\mathcal{C}$*. Then, there exists an access policy* $(\boldsymbol{M}', \rho)$ *such that the following holds:*

- *The matrix* $\boldsymbol{M}' = (M'_{i,j})_{\ell \times s} \in \mathbb{Z}_q^{\ell \times s}$ *satisfies* $M'_{i,j} = 0$ *for all* $(i,j) \in \mathcal{C} \times [s-c]$*.*
- *For any subset* $\mathcal{S} \subset [\ell]$*, if the rows of* $\boldsymbol{M}$ *having indices in* $\mathcal{S}$ *are linearly independent, then so are the rows of* $\boldsymbol{M}'$ *with indices in* $\mathcal{S}$*.*
- *The distribution of the shares* $\{\mu_x\}_{x \in [\ell]}$ *sharing a secret* $z \in \mathbb{Z}_q$ *generated with the matrix* $\boldsymbol{M}$ *is the same as the distribution of the shares* $\{\mu'_x\}_{x \in [\ell]}$ *sharing the same secret* z *generated with the matrix* $\boldsymbol{M}'$*.*

3.3 Decentralized MA-ABE for LSSS

A decentralized multi-authority attribute-based encryption (MA-ABE) system $\mathsf{MA\text{-}ABE} = (\mathsf{GlobalSetup}, \mathsf{AuthSetup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ for access structures captured by linear secret sharing schemes (LSSS) over some finite field $\mathbb{Z}_q$ with $q = q(\lambda)$ consists of five procedures with the following syntax. We denote by $\mathcal{AU}$ the authority universe and by $\mathcal{GID}$ the universe of global identifiers of the users. Additionally, we assume that each authority controls just one attribute, and hence we would use the words 'authority" and 'attribute" interchangeably. This definition naturally generalizes to the situation in which each authority can potentially control an arbitrary number of attributes (see [RW15]).

- $\mathsf{GlobalSetup}(1^\lambda) \mapsto \mathsf{GP}$: The global setup algorithm takes in the security parameter λ in unary 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 $\mathcal{GID}$.

- $\mathsf{AuthSetup}(\mathsf{GP}, u) \mapsto (\mathsf{PK}_u, \mathsf{SK}_u)$: The authority $u \in \mathcal{AU}$ calls the authority setup algorithm during its initialization with the global parameters GP as input and receives back its public and secret key pair $\mathsf{PK}_u, \mathsf{SK}_u$.

- $\mathsf{KeyGen}(\mathsf{GP}, \mathsf{GID}, \mathsf{SK}_u) \mapsto \mathsf{SK}_{\mathsf{GID},u}$: The key generation algorithm takes as input the global parameters GP, a user's global identifier $\mathsf{GID} \in \mathcal{GID}$, and a secret key SK_u of an authority $u \in \mathcal{AU}$. It outputs a secret key $\mathsf{SK}_{\mathsf{GID},u}$ for the user.

- $\mathsf{Enc}(\mathsf{GP}, \mathsf{msg}, (\boldsymbol{M}, \rho), \{\mathsf{PK}_u\}) \mapsto \mathsf{CT}$: The encryption algorithm takes in the global parameters GP, a message msg, an LSSS access policy $(\boldsymbol{M}, \rho)$ such that $\boldsymbol{M}$ is a matrix over $\mathbb{Z}_q$ and ρ is a row-labeling function that assigns to each row of $\boldsymbol{M}$ an attribute/authority in $\mathcal{AU}$, and the set $\{\mathsf{PK}_u\}$ of public keys for all the authorities in the range of ρ. It outputs a ciphertext CT. We assume that the ciphertext implicitly contains $(\boldsymbol{M}, \rho)$.

- $\mathsf{Dec}(\mathsf{GP}, \mathsf{CT}, \{\mathsf{SK}_{\mathsf{GID},u}\}) \mapsto \mathsf{msg}'$: The decryption algorithm takes in the global parameters GP, a ciphertext CT generated with respect to some LSSS access policy $(\boldsymbol{M}, \rho)$, and a collection of keys $\{\mathsf{SK}_{\mathsf{GID},u}\}$ corresponding to user ID-attribute pairs (GID, U) possessed by a user with global identifier GID. It outputs a message msg' when the collection of attributes associated with the secret keys $\{\mathsf{SK}_{\mathsf{GID},u}\}$ satisfies the LSSS access policy $(\boldsymbol{M}, \rho)$, i.e., when the vector $(1, 0, \ldots, 0)$ is contained in the linear span of those rows of $\boldsymbol{M}$ which are mapped by ρ to some attribute/authority $u \in \mathcal{AU}$ such that the secret key $\mathsf{SK}_{\mathsf{GID},u}$ is possessed by the user with global identifier GID. Otherwise, decryption fails.

Correctness: An MA-ABE scheme for LSSS-realizable access structures is said to be *correct* if for every $\lambda \in \mathbb{N}$, every message msg, and $\mathsf{GID} \in \mathcal{GID}$, every LSSS access policy $(\boldsymbol{M}, \rho)$, and

every subset of authorities $U \subseteq \mathcal{AU}$ controlling attributes which satisfy the access structure it holds that

$$\Pr\left[\mathsf{msg}' = \mathsf{msg} \;\middle|\; \begin{array}{c} \mathsf{GP} \leftarrow \mathsf{GlobalSetup}(1^\lambda) \\ \forall u \in U\colon \mathsf{PK}_u, \mathsf{SK}_u \leftarrow \mathsf{AuthSetup}(\mathsf{GP}, u) \\ \forall u \in U\colon \mathsf{SK}_{\mathsf{GID},u} \leftarrow \mathsf{KeyGen}(\mathsf{GP}, \mathsf{GID}, \mathsf{SK}_u) \\ \mathsf{CT} \leftarrow \mathsf{Enc}(\mathsf{GP}, \mathsf{msg}, (\boldsymbol{M}, \rho), \{\mathsf{PK}_u\}) \\ \mathsf{msg}' = \mathsf{Dec}(\mathsf{GP}, \mathsf{CT}, \{\mathsf{SK}_{\mathsf{GID},u}\}_{u \in U}) \end{array}\right] = 1.$$

Static Security: We follow Rouselakis and Waters [RW15] and define static security for MA-ABE systems for LSSS-realizable access structures by the following game between a challenger and an attacker. Here, all queries done by the attacker are sent to the challenger immediately after seeing the global public parameters. We also allow the adversary to corrupt (and thus fully control) a certain set of authorities chosen after seeing the global public parameters and that set of corrupted authorities remains the same until the end of the game.

The game consists of the following phases:

Global setup: The challenger calls $\mathsf{GlobalSetup}(1^\lambda)$ to get and send the global public parameters GP to the attacker.

Adversary's queries: The adversary responds with:

(a) A set $\mathcal{C} \subset \mathcal{AU}$ of corrupt authorities and their respective public keys $\{\mathsf{PK}_u\}_{u \in \mathcal{C}}$, which it might have created in a malicious way.

(b) A set $\mathcal{N} \subset \mathcal{AU}$ of non-corrupt authorities, i.e., $\mathcal{C} \cap \mathcal{N} = \emptyset$, for which it requests the public keys.

(c) A set $\mathcal{Q} = \{(\mathsf{GID}, U)\}$ of secret key queries, where each $\mathsf{GID} \in \mathcal{GID}$ is distinct and each $U \subset \mathcal{N}$.

(d) Two messages $\mathsf{msg}_0, \mathsf{msg}_1 \in \mathbb{G}_T$ of equal length and a challenge LSSS access policy $(\boldsymbol{M}, \rho)$ with ρ labeling each row of $\boldsymbol{M}$ with authorities/attributes in $(\mathcal{C} \cup \mathcal{N})$ subject to the restriction that for each pair $(\mathsf{GID}, U) \in \mathcal{Q}$, the rows of $\boldsymbol{M}$ labeled by authorities/attributes in $(\mathcal{C} \cup U)$ are unauthorized with respect to $(\boldsymbol{M}, \rho)$.

Challenger's replies: The challenger flips a random coin $\beta \leftarrow \{0, 1\}$ and replies with the following:

(a) The public keys $\mathsf{PK}_u \leftarrow \mathsf{AuthSetup}(\mathsf{GP}, u)$ for all $u \in \mathcal{N}$.

(b) The secret keys $\mathsf{SK}_{\mathsf{GID},u} \leftarrow \mathsf{KeyGen}(\mathsf{GP}, \mathsf{GID}, \mathsf{SK}_u)$ for all $(\mathsf{GID}, U) \in \mathcal{Q}$, $u \in U$.

(c) The challenge ciphertext $\mathsf{CT} \leftarrow \mathsf{Enc}(\mathsf{GP}, \mathsf{msg}_\beta, (\boldsymbol{M}, \rho), \{\mathsf{PK}_u\}_{u \in \mathcal{C} \cup \mathcal{N}})$.

Guess: The adversary outputs a guess β' for β.

The advantage of an adversary $\mathcal{A}$ in this game is defined as:

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{MA\text{-}ABE},\mathsf{ST-CPA}}(\lambda) \triangleq \left|\Pr[\beta = \beta'] - 1/2\right|.$$

Definition 3.3 (Static security for MA-ABE for LSSS): A MA-ABE scheme for LSSS-realizable access structures is statically secure if for any PPT adversary $\mathcal{A}$ there exists a negligible function $\mathsf{negl}(\cdot)$ such that for all $\lambda \in \mathbb{N}$, we have $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{MA\text{-}ABE},\mathsf{ST-CPA}}(\lambda) \leq \mathsf{negl}(\lambda)$.

Remark 3.2 (Static security of MA-ABE for LSSS in the Random Oracle Model): Similar to [LW11a, RW15], we additionally consider the aforementioned notion of static 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. In the security proof, we will model H as a random oracle programmed by the challenger. In the security game, therefore, we let the adversary $\mathcal{A}$ submit a collection of H-oracle queries to the challenger immediately after seeing the global public parameters, along with all the other queries it makes in the static security game as described above.

4 Our MA-ABE Scheme from DBDH

In this section, we present our MA-ABE scheme for access structures described by monotone LSSS under the DBDH assumption. As outlined in Remark 4.1, this construction can be tweaked to get an MA-ABE scheme under the CBDH assumption. The scheme is associated with a universe of global identifiers $\mathcal{GID} \subset \{0,1\}^*$, a universe of authority identifiers $\mathcal{AU}$, and we will use the Lewko-Waters [LW11a] transformation to represent the DNF access policies as monotone LSSS. We will assume each authority controls only one attribute in our scheme. However, it can be readily generalized to a scheme where each authority controls an a priori bounded number of attributes using standard techniques [LW11a]. Further, we will assume that all access policies $(\boldsymbol{M}, \rho)$ used in our scheme correspond to a matrix $\boldsymbol{M}$ with at most $s_{\max}$ columns and an injective row-labeling function ρ, i.e., an authority/attribute is associated with at most one row of $\boldsymbol{M}$. Note that following the simple encoding technique devised in [Wat11, LW11a], we can alleviate the injective restriction on the row labeling functions to allow an authority/attribute to appear an a priori bounded number of times within the LSSS access policies.

GlobalSetup(1^λ, $s_{\max}$): The global setup algorithm takes in the security parameter 1^λ encoded in unary and the maximum width of an LSSS matrix supported by the scheme $s_{\max} = s_{\max}(\lambda)$. The procedure runs the bilinear group generator $\mathcal{G}(\lambda)$ to generate a suitable bilinear group $\mathsf{G} = (q, \mathbb{G}, \mathbb{G}_T, g, e)$ of prime order q. The global parameters GP consists of the description of the bilinear group G. Furthermore, we assume a hash function $\mathsf{H}\colon \mathcal{GID} \times [s_{\max}] \to \mathbb{G}$ mapping strings $(\mathsf{GID}, i) \in \mathcal{GID} \times [s_{\max}]$ to elements in $\mathbb{G}$.

AuthSetup(GP, H, u): Given the global parameters GP, the hash function H, and an authority index $u \in \mathcal{AU}$, the algorithm chooses $\alpha_u, y_{u,2}, \ldots, y_{u,s_{\max}} \leftarrow \mathbb{Z}_q$ and outputs

$$\mathsf{PK}_u = (e(g,g)^{\alpha_u}, g^{y_{u,2}}, \ldots, g^{y_{u,s_{\max}}}) \qquad \mathsf{MSK}_u = (\alpha_u, y_{u,2}, \ldots, y_{u,s_{\max}}).$$

KeyGen(GP, H, GID, MSK_u): The key generation algorithm takes as input the global parameters GP, the hash function H, the user's global identifier GID, and the authority's secret key MSK_u. It outputs

$$\mathsf{SK}_{\mathsf{GID},u} = g^{\alpha_u} \prod_{j=2}^{s_{\max}} \mathsf{H}(\mathsf{GID} \parallel j)^{y_{u,j}}.$$

Enc(GP, H, msg, $(\boldsymbol{M}, \rho)$, $\{\mathsf{PK}_u\}$): The encryption algorithm takes as input the global parameters GP, the hash function H, a message $\mathsf{msg} \in \mathbb{G}_T$ to encrypt, an LSSS access structure $(\boldsymbol{M}, \rho)$, where $\boldsymbol{M} = (M_{i,j})_{\ell \times s_{\max}} = (\boldsymbol{M}_1, \ldots, \boldsymbol{M}_\ell)^\top \in \mathbb{Z}_q^{\ell \times s_{\max}}$ and $\rho\colon [\ell] \to \mathcal{AU}$, and public keys of the relevant authorities $\{\mathsf{PK}_u\}$. The function ρ associates rows of $\boldsymbol{M}$ to authorities (recall that we assume that each authority controls a single attribute). We assume that ρ is an injective function, that is, an authority/attribute is associated with at most one row of $\boldsymbol{M}$. The procedure samples

1. $r_1, \ldots, r_\ell \leftarrow \mathbb{Z}_q$,
2. $\boldsymbol{v} = (z, v_2, \ldots, v_{s_{\max}}) \leftarrow \mathbb{Z}_p^{s_{\max}}$, and
3. $\boldsymbol{x} = (x_2, \ldots, x_{s_{\max}}) \leftarrow \mathbb{Z}_p^{s_{\max}-1}$,

and outputs

$$\mathsf{CT} = ((\boldsymbol{M}, \rho), C_0, \{C_{1,i}\}_{i\in[\ell]}, \{C_{2,i}\}_{i\in[\ell]}, \{C_{3,i,j}\}_{i\in[\ell], j\in\{2,\ldots,s_{\max}\}}),$$

where

$$C_0 = e(g,g)^z \mathsf{msg}$$
$$\forall i \in [\ell]\colon\ C_{1,i} = e(g,g)^{\boldsymbol{M}_i \cdot \boldsymbol{v}} e(g,g)^{\alpha_{\rho(i)} r_i}, \quad C_{2,i} = g^{r_i}$$
$$\forall i \in [\ell], j \in \{2,\ldots,s_{\max}\}\colon\ C_{3,i,j} = g^{y_{\rho(i),j} r_i} g^{M_{i,j} x_j}.$$

Dec(GP, H, CT, GID, $\{\mathsf{SK}_{\mathsf{GID},u}\}$): Decryption takes as input the global parameters GP, the hash function H, a ciphertext CT for an access structure $(\boldsymbol{M}, \rho)$ with $\boldsymbol{M} \in \mathbb{Z}_q^{\ell \times s_{\max}}$ and $\rho : [\ell] \to \mathbb{U}$ injective, and the secret keys $\{\mathsf{SK}_{\mathsf{GID},\rho(i)}\}_{i \in I}$ corresponding to a subset of rows of $\boldsymbol{M}$ with indices $I \subseteq [\ell]$. If $(1,0,\ldots,0)$ is *not* in the span of these rows, $\boldsymbol{M}_I$, then decryption fails. Otherwise, the decryptor finds $\{w_i\}_{i \in I}$ such that $(1,0,\ldots,0) = \sum_{i \in I} w_i \boldsymbol{M}_i$. The decryption algorithm, letting $\rho(i)$ for $i \in I$ be the corresponding authority, computes and outputs

$$C_0 \Big/ \prod_{i \in I} \left[\frac{C_{1,i} \cdot \prod_{j=2}^{s_{\max}} e(\mathsf{H}(\mathsf{GID} \parallel j), C_{3,i,j})}{e(\mathsf{SK}_{\mathsf{GID},\rho(i)}, C_{2,i})} \right]^{w_i}.$$

Remark 4.1 (Our MA-ABE Scheme from CBDH): The above construction can be readily modified into one that is provably secure under the CBDH assumption. Let $\mathcal{H}$ be a hardcore bit function for the CBDH problem, that is given a CBDH instance consisting of the group description $\mathsf{G} = (q, \mathbb{G}, \mathbb{G}_T, g, e)$ and three group elements $g^a, g^b, g^c \in \mathbb{G}$ for $a, b, c \leftarrow \mathbb{Z}_q$, the output of $\mathcal{H}$ is defined to be a computationally hard bit in the binary representation of the target $e(g,g)^{abc}$. For the modified construction, let us consider the message space to be $\{0,1\}$ (for simplicity) as opposed to $\mathbb{G}_T$ in the construction above. We modify the component C_0 of the ciphertext as follows. While encrypting a message $\mathsf{msg} \in \{0,1\}$, we compute C_0 as $C_0 = \mathcal{H}(e(g,g)^z) \oplus \mathsf{msg}$ where $\oplus$ denotes addition modulo 2. The other ciphertext components are computed as above. The authority public/secret keys and the user's secret keys are also computed identically to the above scheme.

As for the construction, it is also easy to tweak the security proof of the above MA-ABE scheme in Section 5.2 into a security proof for this modified construction under the intractability of the hardcore bit of the CBDH problem. Specifically, we design an adversary $\mathcal{B}$ which given an instance of the CBDH problem, attempts to compute its hardcore bit by using an adversary $\mathcal{A}$ against the static security of this modified MA-ABE scheme. The reduction algorithm $\mathcal{B}$ proceeds identically to the one described in Section 5.2, except that while generating the challenge ciphertext, it simply sets $C_0 \leftarrow \{0,1\}$. If the guess bit outputted by the adversary at the end of the game matches with C_0, then it outputs the hardcore bit as 0. Otherwise, it outputs the hardcore bit as 1. It is straight-forward to observe that if $\mathcal{A}$ has a non-negligible advantage in guessing the challenge bit, then $\mathcal{B}$ also determines the hardcore bit with non-negligible advantage.

5 Correctness and Security Analysis of Our MA-ABE Scheme

In this section, we provide the correctness and security analysis of the proposed MA-ABE scheme in Section 4.

5.1 Correctness

Assume that the authorities in $\{\mathsf{SK}_{\mathsf{GID},u}\}$ correspond to a qualified set according to the LSSS access structure $(\boldsymbol{M}, \rho)$ associated with CT, that is, the corresponding subset of row indices I corresponds to rows in $\boldsymbol{M}$ that have $(1,0,\ldots,0)$ in their span.

For each $i \in I$, letting $\rho(i)$ be the corresponding authority,

$$\begin{aligned} e\left(\mathsf{SK}_{\mathsf{GID},\rho(i)}, C_{2,i}\right) &= e\left(g^{\alpha_{\rho(i)}} \prod_{j=2}^{s_{\max}} \mathsf{H}(\mathsf{GID} \parallel j)^{y_{\rho(i),j}}, g^{r_i}\right) \\ &= e(g,g)^{\alpha_{\rho(i)} r_i} \prod_{j=2}^{s_{\max}} e\left(\mathsf{H}(\mathsf{GID} \parallel j)^{y_{\rho(i),j}}, g^{r_i}\right) \\ &= e(g,g)^{\alpha_{\rho(i)} r_i} \prod_{j=2}^{s_{\max}} e\left(\mathsf{H}(\mathsf{GID} \parallel j), g\right)^{y_{\rho(i),j} r_i}. \end{aligned}$$

Also, for each $i \in I$,

$$\begin{aligned} \prod_{j=2}^{s_{\max}} e(\mathsf{H}(\mathsf{GID} \parallel j), C_{3,i,j}) &= \prod_{j=2}^{s_{\max}} e(\mathsf{H}(\mathsf{GID} \parallel j), g^{y_{\rho(i),j} r_i} g^{M_{i,j} x_j}) \\ &= \prod_{j=2}^{s_{\max}} e(\mathsf{H}(\mathsf{GID} \parallel j), g)^{y_{\rho(i),j} r_i} e(\mathsf{H}(\mathsf{GID} \parallel j), g)^{M_{i,j} x_j}. \end{aligned}$$

Recall that for each $i \in I$, it holds that $C_{1,i} = e(g,g)^{\boldsymbol{M}_i \cdot \boldsymbol{v}} e(g,g)^{\alpha_{\rho(i)} r_i}$ and so

$$\begin{aligned} &\frac{C_{1,i} \cdot \prod_{j=2}^{s_{\max}} e(\mathsf{H}(\mathsf{GID} \parallel j), C_{3,i,j})}{e(\mathsf{SK}_{\mathsf{GID},\rho(i)}, C_{2,i})} = \\ &\frac{e(g,g)^{\boldsymbol{M}_i \cdot \boldsymbol{v}} e(g,g)^{\alpha_{\rho(i)} r_i} \prod_{j=2}^{s_{\max}} e(\mathsf{H}(\mathsf{GID} \parallel j), g)^{y_{\rho(i),j} r_i} e(\mathsf{H}(\mathsf{GID} \parallel j), g)^{M_{i,j} x_j}}{e(g,g)^{\alpha_{\rho(i)} r_i} \prod_{j=2}^{s_{\max}} e\left(\mathsf{H}(\mathsf{GID} \parallel j), g\right)^{y_{\rho(i),j} r_i}} = \\ &e(g,g)^{\boldsymbol{M}_i \cdot \boldsymbol{v}} \prod_{j=2}^{s_{\max}} e(\mathsf{H}(\mathsf{GID} \parallel j), g)^{M_{i,j} x_j}. \end{aligned}$$

Therefore,

$$\begin{aligned} &\prod_{i \in I} \left[\frac{C_{1,i} \cdot \prod_{j=2}^{s_{\max}} e(\mathsf{H}(\mathsf{GID} \parallel j), C_{3,i,j})}{e(\mathsf{SK}_{\mathsf{GID},\rho(i)}, C_{2,i})} \right]^{w_i} \\ &= \prod_{i \in I} \left[e(g,g)^{\boldsymbol{M}_i \cdot \boldsymbol{v}} \prod_{j=2}^{s_{\max}} e(\mathsf{H}(\mathsf{GID} \parallel j), g)^{M_{i,j} x_j} \right]^{w_i} \\ &= \prod_{i \in I} e(g,g)^{w_i \boldsymbol{M}_i \cdot \boldsymbol{v}} \prod_{j=2}^{s_{\max}} \prod_{i \in I} e(\mathsf{H}(\mathsf{GID} \parallel j), g)^{w_i M_{i,j} x_j} \\ &= e(g,g)^z, \end{aligned}$$

where the last inequality follows since $\sum_{i \in I} w_i \boldsymbol{M}_i \cdot \boldsymbol{v} = (1, 0, \ldots, 0) \cdot \boldsymbol{v} = z$ and $\sum_{i \in I} w_i M_{i,j} = 0$ for every $j \in \{2, \ldots, s_{\max}\}$. Lastly, dividing out this value from C_0 gives the message msg, namely,

$$C_0 / e(g,g)^z = \mathsf{msg}.$$

5.2 Security Analysis

Theorem 5.1: *If the* DBDH *assumption holds, then all* PPT *adversaries have a negligible advantage in breaking the static security of the proposed* MA-ABE *scheme in Section 4 in the random oracle model.*

Proof. Suppose, there exists a PPT adversary, $\mathcal{A}$ that breaks the static security of the proposed MA-ABE scheme in the random oracle model with non-negligible advantage. Using $\mathcal{A}$ as a subroutine, we construct below a PPT adversary $\mathcal{B}$ that has a non-negligible advantage in solving the DBDH problem. The algorithm $\mathcal{B}$ gets an instance of the DBDH problem from its challenger that consists of the group description $\mathsf{G} = (q, \mathbb{G}, \mathbb{G}_T, g, e) \leftarrow \mathcal{G}(1^\lambda)$, three group elements $g^a, g^b, g^c \in \mathbb{G}$ for $a, b, c \leftarrow \mathbb{Z}_q$, and another group element $\tau \in \mathbb{G}_T$ which is either $\tau = e(g,g)^{abc}$ or $\tau \leftarrow \mathbb{G}_T$. The algorithm $\mathcal{B}$ follows:

Generating the Global Public Parameters: $\mathcal{B}$ sets the global public parameters $\mathsf{GP} = \mathsf{G} = (q, \mathbb{G}, \mathbb{G}_T, g, e)$ and invokes the adversary $\mathcal{A}$ against the proposed MA-ABE scheme on input 1^λ and GP.

Attacker's queries: Upon initialization, the adversary $\mathcal{A}$ then sends the following to $\mathcal{B}$:

- A list $\mathcal{C} \subset \mathcal{AU}$ of corrupt authorities and their respective public keys

$$\{\mathsf{PK}_u = (Y_{u,1}, Y_{u,2}, \ldots, Y_{u,s_{\max}})\}_{u \in \mathcal{C}},$$

  where $Y_{u,1} \in \mathbb{G}_T, Y_{u,2}, \ldots, Y_{u,s_{\max}} \in \mathbb{G}$ for all $u \in \mathcal{C}$.
- A set $\mathcal{N} \subset \mathcal{AU}$ of non-corrupt authorities, i.e., $\mathcal{C} \cap \mathcal{N} = \emptyset$, for which $\mathcal{A}$ requests the public keys.
- A set $\mathcal{H} = \{\mathsf{GID}\}$ of distinct GIDs for each of which $\mathcal{A}$ requests the outputs $\{\mathsf{H}(\mathsf{GID}\|2), \ldots, \mathsf{H}(\mathsf{GID}\|s_{\max})\}$ of H (which is modeled as a random oracle in this proof).
- A set $\mathcal{Q} = \{(\mathsf{GID}, U)\}$ of secret key queries, where each $\mathsf{GID} \in \mathcal{GID}$ is distinct and each $U \subset \mathcal{N}$.
- Two messages $\mathsf{msg}_0, \mathsf{msg}_1 \in \mathbb{G}_T$ of equal length, and a challenge LSSS access policy $(\boldsymbol{M}, \rho)$ with $\boldsymbol{M} = (M_{i,j})_{\ell \times s_{\max}} = (\boldsymbol{M}_1, \ldots, \boldsymbol{M}_\ell)^\top \in \mathbb{Z}_q^{\ell \times s_{\max}}$ and $\rho\colon [\ell] \to \mathcal{C} \cup \mathcal{N}$ injective and satisfying the constraint that for each $(\mathsf{GID}, U) \in \mathcal{Q}$, $\rho^{-1}(U \cup \mathcal{C})$ constitutes an unauthorized subset of rows of the access matrix $\boldsymbol{M}$.

Erratum. *The assertion in the following paragraph that replacing $\boldsymbol{M}$ with $\boldsymbol{M}'$ preserves the adversary's view is incorrect and is the central error in our security analysis. Lemma 3.1 is correct as stated and guarantees that the secret shares have the same distribution after applying the matrix transformation. However, it does not imply that the ciphertext components containing the individual products $M_{i,j}x_j$ have the same distribution after this replacement.*

Before answering the above queries from $\mathcal{A}$, $\mathcal{B}$ substitutes the secret sharing matrix $\boldsymbol{M}$ with the matrix $\boldsymbol{M}'$ from Lemma 3.1 computed using $\rho^{-1}(\mathcal{C})$ as the unauthorized subset of rows. By the guarantee of Lemma 3.1, from now on, if $\mathcal{B}$ uses $\boldsymbol{M}'$ instead of $\boldsymbol{M}$ in the simulation, the view of $\mathcal{A}$ in the simulated game is information theoretically the same as if $\mathcal{B}$ would have used the original matrix $\boldsymbol{M}$. Furthermore, Lemma 3.1 implies that if we assume the subspace spanned by $\boldsymbol{M}_{\rho^{-1}(\mathcal{C})}$ has dimension $\tilde{c}$, then so is the dimension of the subspace spanned by $\boldsymbol{M}'_{\rho^{-1}(\mathcal{C})}$ and $M'_{i,j} = 0$ for all $(i,j) \in \rho^{-1}(\mathcal{C}) \times [s_{\max} - \tilde{c}]$. $\mathcal{B}$ then proceeds to compute the reply to $\mathcal{A}$. Denote $\hat{s}_{\max} = s_{\max} - \tilde{c}$, where $\tilde{c}$ is the dimension of the subspace spanned by the rows of $\boldsymbol{M}_{\rho^{-1}(\mathcal{C})}$, the latter being the rows of $\boldsymbol{M}$ controlled by corrupted authorities, $\mathcal{C}$.

Generating authority public keys: There are two cases to consider:

1. **Case 1** — $u \in \mathcal{N} \setminus \rho([\ell])$ (the attribute owned by the authority is not present within the challenge access structure) — In this case, $\mathcal{B}$ executes $\mathsf{AuthSetup}$ according to the scheme. That is, $\mathcal{B}$ picks $\alpha_u, y_{u,2}, \ldots, y_{u,s_{\max}} \leftarrow \mathbb{Z}_q$ itself, and outputs the public key

$$\mathsf{PK}_u = (e(g,g)^{\alpha_u}, g^{y_{u,2}}, \ldots, g^{y_{u,s_{\max}}}).$$

2. **Case 2** — $u \in \rho([\ell]) \setminus \mathcal{C}$ (the attribute owned by the authority appears in the challenge access structure) — In this case, $\mathcal{B}$ picks $\alpha'_u, y'_{u,2}, \ldots, y'_{u,\hat{s}_{\max}}, y_{u,\hat{s}_{\max}+1}, \ldots, y_{u,s_{\max}} \leftarrow \mathbb{Z}_q$, implicitly sets $\alpha_u = \alpha'_u + abM'_{\rho^{-1}(u),1}$ and $y_{u,j} = y'_{u,j} + aM'_{\rho^{-1}(u),j}$ for $j \in \{2, \ldots, \hat{s}_{\max}\}$ (recall that ρ is injective so these are well-defined), and outputs the public key

$$\begin{aligned}\mathsf{PK}_u &= \left(e(g,g)^{\alpha_u}, \{g^{y_{u,j}}\}_{j\in\{2,\ldots,\hat{s}_{\max}\}}, \{g^{y_{u,j}}\}_{j\in\{\hat{s}_{\max}+1,\ldots,s_{\max}\}}\right) \\ &= \left(e(g,g)^{\alpha'_u} \cdot e(g^a, g^b)^{M'_{\rho^{-1}(u),1}}, \left\{g^{y'_{u,j}} \cdot (g^a)^{M'_{\rho^{-1}(u),j}}\right\}_{j\in\{2,\ldots,\hat{s}_{\max}\}},\right. \\ &\qquad \left.\{g^{y_{u,j}}\}_{j\in\{\hat{s}_{\max}+1,\ldots,s_{\max}\}}\right).\end{aligned}$$

   Notice that α_u and $\{y_{u,j}\}_{j\in\{2,\ldots,\hat{s}_{\max}\}}$ are distributed uniformly over $\mathbb{Z}_q$ and so each component in PK_u is properly distributed.

Answering H oracle queries: There are two cases to consider:

1. **Case 1** — $(\mathsf{GID}, U) \in \mathcal{Q}$ and $U \cap \rho([\ell]) \neq \emptyset$ — In this case, according to the game restriction, the rows having indices in $\rho^{-1}(U \cup \mathcal{C})$ constitutes an unauthorized subset of rows of $\boldsymbol{M}$ and hence, that of $\boldsymbol{M'}$ by Lemma 3.1. Therefore, there exists a vector $\boldsymbol{d} = (d_1, \ldots, d_{s_{\max}}) \in \mathbb{Z}_q^{s_{\max}}$ such that $d_1 = 1$ and the inner product of $\boldsymbol{d}$ with any of the rows of $\boldsymbol{M'}$ having indices in $\rho^{-1}(U \cup \mathcal{C})$ is zero. Additionally, since the rows of $\boldsymbol{M'}$ having indices in $\rho^{-1}(\mathcal{C})$, each of which has dimension $s_{\max}$, together span a subspace of dimension c and $M'_{i,j} = 0$ for all $(i,j) \in \rho^{-1}(\mathcal{C}) \times [\hat{s}_{\max}]$, it follows that the rows of $\boldsymbol{M'}$ having indices in $\rho^{-1}(\mathcal{C})$ spans the entire subspace $\mathbb{V} = \left\{ (\overbrace{0, \ldots, 0}^{\hat{s}_{\max}}, \boldsymbol{u}) \,\middle|\, \boldsymbol{u} \in \mathbb{Z}_q^c \right\}$. Thus, it follows that $\boldsymbol{d}$ is orthogonal to any of the vectors

$$\left\{ (\overbrace{0, \ldots, 0}^{\hat{s}_{\max}}, \overbrace{0, \ldots, 0}^{j-1}, 1, \overbrace{0, \ldots, 0}^{s_{\max}-j}) \right\}_{j\in\{\hat{s}_{\max}+1,\ldots,s_{\max}\}}.$$

   In other words, $d_j = 0$ for all $j \in \{\hat{s}_{\max}+1, \ldots, s_{\max}\}$. Combining the above two facts, we have $(\boldsymbol{M}'_i|_{[\hat{s}_{\max}]}) \cdot (\boldsymbol{d}|_{[\hat{s}_{\max}]}) = 0$ for all $i \in \rho^{-1}(U)$, where for a vector $\boldsymbol{z}$, $\boldsymbol{z}|_X$ denotes a vector formed by taking the entries of $\boldsymbol{z}$ having indices in the set $X \subset \mathbb{N}$. For simplicity of notations, let us denote $\boldsymbol{M}'_i \star \boldsymbol{d} = (\boldsymbol{M}'_i|_{[\hat{s}_{\max}]}) \cdot (\boldsymbol{d}|_{[\hat{s}_{\max}]})$ for all $i \in \rho^{-1}(U)$. In this case, $\mathcal{B}$ samples $h'_2, \ldots, h'_{\hat{s}_{\max}}, h_{\hat{s}_{\max}+1}, \ldots, h_{s_{\max}} \leftarrow \mathbb{Z}_q$, and outputs

$$\left\{\mathsf{H}(\mathsf{GID}\|j) = (g^b)^{d_j} g^{h'_j}\right\}_{j\in\{2,\ldots,\hat{s}_{\max}\}},$$
$$\left\{\mathsf{H}(\mathsf{GID}\|j) = g^{h_j}\right\}_{j\in\{\hat{s}_{\max}+1,\ldots,s_{\max}\}}.$$

2. **Case 2** — $(\mathsf{GID}, U) \notin \mathcal{Q}$ or $U \cap \rho([\ell]) = \emptyset$ — In this case, $\mathcal{B}$ samples $\mathsf{H}(\mathsf{GID}\|j) \leftarrow \mathbb{G}$ for all $j \in \{2, \ldots, s_{\max}\}$ and outputs them.

All the outputs of the random oracle H as programmed by $\mathcal{B}$ are clearly uniformly distributed in $\mathbb{G}$.

Generating secret keys: For a query $(\mathsf{GID}, U) \in \mathcal{Q}$, the simulator $\mathcal{B}$ has to create a secret key $\mathsf{SK}_{\mathsf{GID},u}$ for every $u \in U$. There are two cases to consider:

1. **Case 1** — $u \in U \setminus \rho([\ell])$ (that is, the authority is not present in the challenge policy) — In this case, $\mathcal{B}$ executes KeyGen according to the scheme. More precisely, in this case, $\mathcal{B}$ knows $\alpha_u, y_{u,2}, \ldots, y_{u,s_{\max}}$, and hence can compute

$$\mathsf{SK}_{\mathsf{GID},u} = g^{\alpha_u} \prod_{j=2}^{s_{\max}} \mathsf{H}(\mathsf{GID}\|j)^{y_{u,j}},$$

   where $\{\mathsf{H}(\mathsf{GID}\|j)\}_{j\in\{2,\ldots,s_{\max}\}}$ are simulated using the strategy described above.
2. **Case 2** — $u \in U \cap \rho([\ell])$ — In this case, $\mathcal{B}$ outputs:

$$\mathsf{SK}_{\mathsf{GID},u} = g^{\alpha_u} \prod_{j=2}^{s_{\max}} \mathsf{H}(\mathsf{GID}\|j)^{y_{u,j}},$$

   which can be computed as (recalling that $\boldsymbol{M}'_{\rho^{-1}(u)} \star \boldsymbol{d} = (\boldsymbol{M}'_{\rho^{-1}(u)}|_{[\hat{s}_{\max}]}) \cdot (\boldsymbol{d}|_{[\hat{s}_{\max}]}) = 0$):

$$\begin{aligned}
&g^{\alpha'_u} \prod_{j=2}^{\hat{s}_{\max}} \left((g^b)^{d_j y'_{u,j}} (g^a)^{h'_j M'_{\rho^{-1}(u),j}} g^{h'_j y'_{u,j}} \right) \prod_{j=\hat{s}_{\max}+1}^{s_{\max}} g^{h_j y_{u,j}} \\
&= g^{ab(\boldsymbol{M}'_{\rho^{-1}(u)} \star \boldsymbol{d})} g^{\alpha'_u} \prod_{j=2}^{\hat{s}_{\max}} \left((g^b)^{d_j y'_{u,j}} (g^a)^{h'_j M'_{\rho^{-1}(u),j}} g^{h'_j y'_{u,j}} \right) \prod_{j=\hat{s}_{\max}+1}^{s_{\max}} g^{h_j y_{u,j}} \\
&= g^{abM'_{\rho^{-1}(u),1} + \alpha'_u} \prod_{j=2}^{\hat{s}_{\max}} g^{(bd_j + h'_j)(aM'_{\rho^{-1}(u),j} + y'_{u,j})} \prod_{j=\hat{s}_{\max}+1}^{s_{\max}} g^{h_j y_{u,j}} \\
&= g^{\alpha_u} \prod_{j=2}^{s_{\max}} \mathsf{H}(\mathsf{GID}\|j)^{y_{u,j}},
\end{aligned}$$

   where the first equality holds since $\boldsymbol{M}'_{\rho^{-1}(u)} \star \boldsymbol{d} = 0$ and the second equality holds since $d_1 = 1$. The correctness of the last step follows thanks to the simulation strategies for the authority public key for the authority $u \in \rho([\ell]) \setminus \mathcal{C}$ (that is, due to the setting of $\alpha_u, \{y_{u,j}\}_{j\in\{2,\ldots,\hat{s}_{\max}\}}$ as $\alpha_u = \alpha'_u + abM'_{\rho^{-1}(u),1}$ and $y_{u,j} = y'_{u,j} + aM'_{\rho^{-1}(u),j}$ for $j \in \{2, \ldots, \hat{s}_{\max}\}$) and the setting of the outputs of the H oracle for the global identity GID with $U \cap \rho([\ell]) \neq \emptyset$ described above.

$\mathcal{B}$ outputs $\{\mathsf{SK}_{\mathsf{GID},u}\}_{u\in U}$.

Generating the challenge ciphertext: $\mathcal{B}$ implicitly sets

$$\boldsymbol{v} = (z, v_2, \ldots, v_{s_{\max}}) = (-abc, 0, \ldots, 0) \in \mathbb{Z}_q^{s_{\max}}$$

and

$$\boldsymbol{x} = (x_2, \ldots, x_{s_{\max}}) = (\overbrace{-ac, \ldots, -ac}^{\{2,\ldots,\hat{s}_{\max}\}}, \overbrace{0, \ldots, 0}^{\{\hat{s}_{\max}+1,\ldots,s_{\max}\}}) \in \mathbb{Z}_q^{s_{\max}-1}.$$

For each $i \in [\ell]$, we consider the following two cases:

1. **Case 1** — $\rho(i) \in \mathcal{C}$ — This means the attribute associated with this row corresponds to a corrupt authority. In this case, it holds that $\boldsymbol{M}'_i \cdot \boldsymbol{v} = 0$ and $M'_{i,j} x_j = 0$ for all $j \in \{2, \ldots, s_{\max}\}$ due to the above implicit setting of $\boldsymbol{v}, \boldsymbol{x}$ and the fact that $\boldsymbol{M}'_i|_{[\hat{s}_{\max}]} =$

$\left\{\overbrace{0,\ldots,0}^{[s_{\max}]}\right\}$. Thus, for each such row, $\mathcal{B}$ picks $r_i \leftarrow \mathbb{Z}_q$, and using the authority public key $\mathsf{PK}_{\rho(i)} = (Y_{\rho(i),1}, Y_{\rho(i),2}, \ldots, Y_{\rho(i),s_{\max}})$ obtained from $\mathcal{A}$ it computes

$$C_{1,i} = Y_{\rho(i),1}^{r_i} = e(g,g)^{\boldsymbol{M}_i' \cdot \boldsymbol{v}} Y_{\rho(i),1}^{r_i}, \quad C_{2,i} = g^{r_i},$$
$$\forall j \in \{2,\ldots,s_{\max}\}\colon C_{3,i,j} = Y_{\rho(i),j}^{r_i} = Y_{\rho(i),j}^{r_i} g^{M_{i,j}' x_j}.$$

2. **Case 2** — $\rho(i) \in \mathcal{N}$ — This means the authority associated with this row is uncorrupted. In this case, we have $\boldsymbol{M}_i' \cdot \boldsymbol{v} = -abcM_{i,1}'$ and $M_{i,j}' x_j = -acM_{i,j}'$ or 0 depending on whether $j \in \{2,\ldots,\hat{s}_{\max}\}$ or $j \in \{\hat{s}_{\max}+1,\ldots,s_{\max}\}$. Also for each such row, we have $\alpha_{\rho(i)} = \alpha'_{\rho(i)} + abM_{i,1}'$ and $y_{\rho(i),j} = y'_{\rho(i),j} + aM_{i,j}'$ for $j \in \{2,\ldots,\hat{s}_{\max}\}$ while $\mathcal{B}$ explicitly knows $y_{\rho(i),j}$ for $j \in \{\hat{s}_{\max}+1,\ldots,s_{\max}\}$. Hence, for each such row $\mathcal{B}$ implicitly sets $r_i = c$ and computes

$$\begin{aligned}
C_{1,i} &= e(g^c,g)^{\alpha'_{\rho(i)}} = e(g,g)^{-abcM_{i,1}'} e(g,g)^{(\alpha'_{\rho(i)} + abM_{i,1}')c} \\
&= e(g,g)^{\boldsymbol{M}_i' \cdot \boldsymbol{v}} e(g,g)^{\alpha_{\rho(i)} r_i}, \\
C_{2,i} &= g^c = g^{r_i}, \\
\forall j \in \{2,\ldots,\hat{s}_{\max}\}\colon C_{3,i,j} &= (g^c)^{y'_{\rho(i),j}} = g^{(y'_{\rho(i),j} + aM_{i,j}')c} g^{M_{i,j}'(-ac)} \\
&= g^{y_{\rho(i),j} r_i} g^{M_{i,j}' x_j}, \\
\forall j \in \{\hat{s}_{\max}+1,\ldots,s_{\max}\}\colon C_{3,i,j} &= (g^c)^{y_{\rho(i),j}} = g^{y_{\rho(i),j} r_i} g^{M_{i,j}' x_j}
\end{aligned}$$

Also, $\mathcal{B}$ sets $C_0 = \mathsf{msg}_\beta \cdot \tau$, where $\beta \leftarrow \{0,1\}$ and τ is the challenge term of the given DBDH instance. Finally, since $\boldsymbol{v}$, $\boldsymbol{x}$, and $\{r_i\}_{i\in[\ell]}$ are not properly distributed, $\mathcal{B}$ re-randomizes the ciphertext using the algorithm CTRandomize described below.

▪ Ciphertext Re-Randomizing Algorithms

The algorithm described below provides properly distributed ciphertexts even if the randomness used within the ciphertexts inputted into the algorithm are not uniform. The algorithm uses only publicly available information to perform the re-randomization. This algorithm is used to rectify the distribution of the challenge ciphertext in our reduction.

$\mathsf{CTRandomize}\left(\mathsf{GP}, (\boldsymbol{M},\rho), \mathsf{CT}, \{\mathsf{PK}_u\}_{u\in\rho([\ell])}\right)$: The input of the procedure consists of the global parameters GP, an LSSS access policy $(\boldsymbol{M},\rho)$, where $\boldsymbol{M} = (M_{i,j})_{\ell \times s_{\max}} = (\boldsymbol{M}_1,\ldots,\boldsymbol{M}_\ell)^\top \in \mathbb{Z}_q^{\ell \times s_{\max}}$ and $\rho\colon [\ell] \to \mathcal{AU}$, a ciphertext $\mathsf{CT} = ((\boldsymbol{M},\rho), C_0, \{C_{1,i}\}_{i\in[\ell]}, \{C_{2,i}\}_{i\in[\ell]}, \{C_{3,i,j}\}_{i\in[\ell],j\in\{2,\ldots,s_{\max}\}})$, and a set of public keys $\{\mathsf{PK}_u\}_{u\in\rho([\ell])}$ such that $\rho([\ell]) \subseteq \mathcal{AU}$. The procedure operates as follows:

1. Sample
   (a) $r_1',\ldots,r_\ell' \leftarrow \mathbb{Z}_q$,
   (b) $\boldsymbol{v}' = (z', v_2',\ldots,v_{s_{\max}}') \leftarrow \mathbb{Z}_q^{s_{\max}}$,
   (c) $\boldsymbol{x}' = (x_2',\ldots,x_{s_{\max}}') \leftarrow \mathbb{Z}_q^{s_{\max}-1}$.
2. Output

$$\mathsf{CT}' = \begin{pmatrix} C_0' = C_0 e(g,g)^{z'}, \\ \forall i \in [\ell]\colon C_{1,i}' = C_{1,i} e(g,g)^{\boldsymbol{M}_i \cdot \boldsymbol{v}'} e(g,g)^{\alpha_{\rho(i)} r_i'}, \quad C_{2,i}' = C_{2,i} g^{r_i'}, \\ \forall i \in [\ell], j \in \{2,\ldots,s_{\max}\}\colon C_{3,i,j}' = C_{3,i,j} g^{y_{\rho(i),j} r_i'} g^{M_{i,j} x_j'} \end{pmatrix}.$$

Guess: If $\mathcal{A}$ guesses the challenge bit $\beta \in \{0,1\}$ correctly, $\mathcal{B}$ outputs 1. Otherwise, if $\mathcal{A}$ guesses the bit β wrongly, $\mathcal{B}$ outputs 0. To see why this finishes the reduction, observe that when τ is uniformly random in $\mathbb{G}_T$, from $\mathcal{A}$'s point of view, there is no information about the challenge bit β in the challenge ciphertext and so the probability of outputting $\beta' = \beta$ is exactly $1/2$. On the other hand, when τ is the DBDH value $e(g,g)^{abc}$, $\mathcal{A}$ outputs $\beta' = \beta$ with probability $1/2 + \epsilon(\lambda)$, where $\epsilon(\lambda)$ is the advantage of $\mathcal{A}$ in the static security game for our MA-ABE scheme—this is true since all the keys are distributed correctly and the ciphertext is a well distributed encryption of msg_β where $\beta \leftarrow \{0,1\}$ and so $\mathcal{A}$ outputs $\beta' = \beta$ with probability $1/2 + \epsilon(\lambda)$. Hence, if $\mathcal{A}$ has a non-negligible advantage against the proposed MA-ABE scheme in the static security game, so has $\mathcal{B}$ in guessing the DBDH challenge. □

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, *Conference on Computer and Communications Security - CCS 2017*, pages 647–664. ACM, 2017.

AC16. Shashank Agrawal and Melissa Chase. A study of pair encodings: Predicate encryption in prime order groups. In Eyal Kushilevitz and Tal Malkin, editors, *Theory of Cryptography - 13th International Conference, TCC 2016-A, Tel Aviv, Israel, January 10-13, 2016, Proceedings, Part II*, volume 9563 of *Lecture Notes in Computer Science*, pages 259–288. Springer, 2016.

AC17a. Shashank Agrawal and Melissa Chase. FAME: fast attribute-based message encryption. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, pages 665–682. ACM, 2017.

AC17b. Shashank Agrawal and Melissa Chase. Simplifying design and analysis of complex predicate encryption schemes. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part I*, volume 10210 of *Lecture Notes in Computer Science*, pages 627–656. Springer, 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, *Advances in Cryptology - ASIACRYPT 2011*, volume 7073 of *Lecture Notes in Computer Science*, pages 21–40. Springer, 2011.

ALdP11. Nuttapong Attrapadung, Benoît Libert, and Elie de Panafieu. Expressive key-policy attribute-based encryption with constant-size ciphertexts. In Dario Catalano, Nelly Fazio, Rosario Gennaro, and Antonio Nicolosi, editors, *Public Key Cryptography - PKC 2011 - 14th International Conference on Practice and Theory in Public Key Cryptography, Taormina, Italy, March 6-9, 2011. Proceedings*, volume 6571 of *Lecture Notes in Computer Science*, pages 90–108. Springer, 2011.

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, *Advances in Cryptology - CRYPTO 2019*, volume 11693 of *Lecture Notes in Computer Science*, pages 765–797. Springer, 2019.

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, *Advances in Cryptology - EUROCRYPT 2014*, volume 8441 of *Lecture Notes in Computer Science*, pages 557–577. Springer, 2014.

Att16. Nuttapong Attrapadung. Dual system encryption framework in prime-order groups via computational pair encodings. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology - ASIACRYPT 2016*, volume 10032 of *Lecture Notes in Computer Science*, pages 591–623. Springer, 2016.

Att19. Nuttapong Attrapadung. Unbounded dynamic predicate compositions in attribute-based encryption. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2019*, volume 11476 of *Lecture Notes in Computer Science*, pages 34–67. Springer, 2019.

AWY20. Shweta Agrawal, Daniel Wichs, and Shota Yamada. Optimal broadcast encryption from LWE and pairings in the standard model. 2020. `https://eprint.iacr.org/2020/1179`.

AY20. Shweta Agrawal and Shota Yamada. Optimal broadcast encryption from pairings and LWE. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology - EUROCRYPT 2020*, volume 12105 of *Lecture Notes in Computer Science*, pages 13–43. Springer, 2020.

BBS04. Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In Matthew K. Franklin, editor, *Advances in Cryptology - CRYPTO 2004, 24th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 2004, Proceedings*, volume 3152 of *Lecture Notes in Computer Science*, pages 41–55. Springer, 2004.

BF01. Dan Boneh and Matthew K. Franklin. Identity-based encryption from the weil pairing. In Joe Kilian, editor, *Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001, Proceedings*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–229. Springer, 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, *Advances in Cryptology - EUROCRYPT 2014*, volume 8441 of *Lecture Notes in Computer Science*, pages 533–556. Springer, 2014.

BL88. Josh Cohen Benaloh and Jerry Leichter. Generalized secret sharing and monotone functions. In Shafi Goldwasser, editor, *Advances in Cryptology - CRYPTO 1988*, volume 403 of *Lecture Notes in Computer Science*, pages 27–35. Springer, 1988.

Boy13. Xavier Boyen. Attribute-based functional encryption on lattices. In Amit Sahai, editor, *Theory of Cryptography Conference - TCC 2013*, volume 7785 of *Lecture Notes in Computer Science*, pages 122–142. Springer, 2013.

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

BV16. Zvika Brakerski and Vinod Vaikuntanathan. Circuit-ABE from LWE: unbounded attributes and semi-adaptive security. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology - CRYPTO 2016*, volume 9816 of *Lecture Notes in Computer Science*, pages 363–384. Springer, 2016.

BV20. Zvika Brakerski and Vinod Vaikuntanathan. Lattice-inspired broadcast encryption and succinct ciphertext-policy ABE. 2020. `https://eprint.iacr.org/2020/191`.

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, *Conference on Computer and Communications Security - CCS 2009*, pages 121–130. ACM, 2009.

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, *Advances in Cryptology - EUROCRYPT 2018*, volume 10820 of *Lecture Notes in Computer Science*, pages 503–534. Springer, 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, *Advances in Cryptology - EUROCRYPT 2015*, volume 9057 of *Lecture Notes in Computer Science*, pages 595–624. Springer, 2015.

Cha07. Melissa Chase. Multi-authority attribute based encryption. In Salil P. Vadhan, editor, *Theory of Cryptography Conference - TCC 2007*, volume 4392 of *Lecture Notes in Computer Science*, pages 515–534. Springer, 2007.

Che06. Jung Hee Cheon. Security analysis of the strong diffie-hellman problem. In Serge Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings*, volume 4004 of *Lecture Notes in Computer Science*, pages 1–11. Springer, 2006.

CMM16. Melissa Chase, Mary Maller, and Sarah Meiklejohn. Déjà Q all over again: Tighter and broader reductions of q-type assumptions. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part II*, volume 10032 of *Lecture Notes in Computer Science*, pages 655–681. Springer, 2016.

CW14. Jie Chen and Hoeteck Wee. Semi-adaptive attribute-based encryption and improved delegation for boolean formula. In Michel Abdalla and Roberto De Prisco, editors, *Security and Cryptography for Networks - 9th International Conference, SCN 2014, Amalfi, Italy, September 3-5, 2014. Proceedings*, volume 8642 of *Lecture Notes in Computer Science*, pages 277–297. Springer, 2014.

DDM15. Pratish Datta, Ratna Dutta, and Sourav Mukhopadhyay. Compact attribute-based encryption and signcryption for general circuits from multilinear maps. In Alex Biryukov and Vipul Goyal, editors, *Progress in Cryptology - INDOCRYPT 2015 - 16th International Conference on Cryptology in India, Bangalore, India, December 6-9, 2015, Proceedings*, volume 9462 of *Lecture Notes in Computer Science*, pages 3–24. Springer, 2015.

DH76. Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions of Information Theory*, 22(6):644–654, 1976.

DKW20. Pratish Datta, Ilan Komargodski, and Brent Waters. Decentralized multi-authority ABE for DNFs from LWE. Cryptology ePrint Archive, Report 2020/1386, 2020. `https://eprint.iacr.org/2020/1386`.

DKW21. Pratish Datta, Ilan Komargodski, and Brent Waters. Decentralized multi-authority ABE for dnfs from LWE. In Anne Canteaut and François-Xavier Standaert, editors, *Advances in Cryptology -*

EUROCRYPT 2021 - 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17-21, 2021, Proceedings, Part I, volume 12696 of *Lecture Notes in Computer Science*, pages 177–209. Springer, 2021.

Fre10. David Mandell Freeman. Converting pairing-based cryptosystems from composite-order groups to prime-order groups. In Henri Gilbert, editor, *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Monaco / French Riviera, May 30 - June 3, 2010. Proceedings*, volume 6110 of *Lecture Notes in Computer Science*, pages 44–61. Springer, 2010.

Gam85. Taher El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions of Information Theory*, 31(4):469–472, 1985.

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, *Advances in Cryptology - CRYPTO 2013*, volume 8043 of *Lecture Notes in Computer Science*, pages 479–499. Springer, 2013.

GKW17. Rishab Goyal, Venkata Koppula, and Brent Waters. Lockable obfuscation. In Chris Umans, editor, *Symposium on Foundations of Computer Science - FOCS 2017*, pages 612–621. IEEE Computer Society, 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, *Conference on Computer and Communications Security - CCS 2006*, pages 89–98. ACM, 2006.

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, *Applied Cryptography and Network Security - 11th International Conference, ACNS 2013, Banff, AB, Canada, June 25-28, 2013. Proceedings*, volume 7954 of *Lecture Notes in Computer Science*, pages 357–372. Springer, 2013.

GVW13. Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Attribute-based encryption for circuits. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *Symposium on Theory of Computing - STOC 2013*, pages 545–554. ACM, 2013.

GW20. Junqing Gong and Hoeteck Wee. Adaptively secure ABE for DFA from k-Lin and more. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology - EUROCRYPT 2020*, volume 12107 of *Lecture Notes in Computer Science*, pages 278–308. Springer, 2020.

GWW19. Junqing Gong, Brent Waters, and Hoeteck Wee. ABE for DFA from k-Lin. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology - CRYPTO 2019*, volume 11693 of *Lecture Notes in Computer Science*, pages 732–764. Springer, 2019.

Jou04. Antoine Joux. A one round protocol for tripartite diffie-hellman. *Journal of Cryptology*, 17(4):263–276, 2004.

Kim19. Sam Kim. Multi-authority attribute-based encryption from LWE in the OT model. 2019. `https://eprint.iacr.org/2019/280`.

KL15. Lucas Kowalczyk and Allison Bishop Lewko. Bilinear entropy expansion from the decisional linear assumption. In Rosario Gennaro and Matthew Robshaw, editors, *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part II*, volume 9216 of *Lecture Notes in Computer Science*, pages 524–541. Springer, 2015.

KOS01. Masao Kasahara, Kiyoshi Ogishi, and Ryuichi Sakai. Cryptosystems based on pairings. In *SCIS 2001, Osio, Japan*, 2001.

KW19. Lucas Kowalczyk and Hoeteck Wee. Compact adaptively secure ABE for NC^1 from k-Lin. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2019*, volume 11476 of *Lecture Notes in Computer Science*, pages 3–33. Springer, 2019.

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, *Progress in Cryptology - INDOCRYPT 2008*, volume 5365 of *Lecture Notes in Computer Science*, pages 426–436. Springer, 2008.

Len01. Arjen K. Lenstra. Unbelievable security. matching AES security using public key systems. In Colin Boyd, editor, *Advances in Cryptology - ASIACRYPT 2001, 7th International Conference on the Theory and Application of Cryptology and Information Security, Gold Coast, Australia, December 9-13, 2001, Proceedings*, volume 2248 of *Lecture Notes in Computer Science*, pages 67–86. Springer, 2001.

Lew12. Allison B. Lewko. Tools for simulating features of composite order bilinear groups in the prime order setting. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings*, volume 7237 of *Lecture Notes in Computer Science*, pages 318–335. Springer, 2012.

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, *Advances in Cryptology - EUROCRYPT 2020*, volume 12107 of *Lecture Notes in Computer Science*, pages 247–277. Springer, 2020.

LL20b. Huijia Lin and Ji Luo. Succinct and adaptively secure ABE for arithmetic branching programs from k-Lin. 2020. https://eprint.iacr.org/2020/1139.

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, *Advances in Cryptology - EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 62–91. Springer, 2010.

LW10. Allison B. Lewko and Brent Waters. New techniques for dual system encryption and fully secure HIBE with short ciphertexts. In Daniele Micciancio, editor, *Theory of Cryptography Conference - TCC 2010*, volume 5978 of *Lecture Notes in Computer Science*, pages 455–479. Springer, 2010.

LW11a. Allison B. Lewko and Brent Waters. Decentralizing attribute-based encryption. In Kenneth G. Paterson, editor, *Advances in Cryptology - EUROCRYPT 2011*, volume 6632 of *Lecture Notes in Computer Science*, pages 568–588. Springer, 2011.

LW11b. Allison B. Lewko and Brent Waters. Unbounded HIBE and attribute-based encryption. In Kenneth G. Paterson, editor, *Advances in Cryptology - EUROCRYPT 2011*, volume 6632 of *Lecture Notes in Computer Science*, pages 547–567. Springer, 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, *Advances in Cryptology - CRYPTO 2012*, volume 7417 of *Lecture Notes in Computer Science*, pages 180–198. Springer, 2012.

MKE08. Sascha Müller, Stefan Katzenbeisser, and Claudia Eckert. Distributed attribute-based encryption. In Pil Joong Lee and Jung Hee Cheon, editors, *International Conference on Information Security and Cryptology - ICISC 2008*, volume 5461 of *Lecture Notes in Computer Science*, pages 20–36. Springer, 2008.

MKE09. Sascha Müller, Stefan Katzenbeisser, and Claudia Eckert. On multi-authority ciphertext-policy attribute-based encryption. *Bulletin of the Korean Mathematical Society*, 46:803–819, 07 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, *Conference on Computer and Communications Security - CCS 2007*, pages 195–203. ACM, 2007.

OT10. Tatsuaki Okamoto and Katsuyuki Takashima. Fully secure functional encryption with general relations from the decisional linear assumption. In Tal Rabin, editor, *Advances in Cryptology - CRYPTO 2010*, volume 6223 of *Lecture Notes in Computer Science*, pages 191–208. Springer, 2010.

OT12. Tatsuaki Okamoto and Katsuyuki Takashima. Fully secure unbounded inner-product and attribute-based encryption. In Xiaoyun Wang and Kazue Sako, editors, *Advances in Cryptology - ASIACRYPT 2012*, volume 7658 of *Lecture Notes in Computer Science*, pages 349–366. Springer, 2012.

OT20. Tatsuaki Okamoto and Katsuyuki Takashima. Decentralized attribute-based encryption and signatures. *IEICE Transactions on Fundamentals of Electronics, Communications, and Computer Sciences*, 103-A(1):41–73, 2020.

Reg05. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *Symposium on Theory of Computing - STOC 2005*, pages 84–93. ACM, 2005.

RSA78. Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

RW15. Yannis Rouselakis and Brent Waters. Efficient statically-secure large-universe multi-authority attribute-based encryption. In *International Conference on Financial Cryptography and Data Security*, pages 315–332, 2015.

SHI+12. Yumi Sakemi, Goichiro Hanaoka, Tetsuya Izu, Masahiko Takenaka, and Masaya Yasuda. Solving a discrete logarithm problem with auxiliary input on a 160-bit elliptic curve. In Marc Fischlin, Johannes A. Buchmann, and Mark Manulis, editors, *Public Key Cryptography - PKC 2012 - 15th International Conference on Practice and Theory in Public Key Cryptography, Darmstadt, Germany, May 21-23, 2012. Proceedings*, volume 7293 of *Lecture Notes in Computer Science*, pages 595–608. Springer, 2012.

SW05. Amit Sahai and Brent Waters. Fuzzy identity-based encryption. In Ronald Cramer, editor, *Advances in Cryptology - EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 457–473. Springer, 2005.

TA20. Junichi Tomida and Nuttapong Attrapadung. Unbounded dynamic predicate compositions in ABE from standard assumptions. 2020. https://eprint.iacr.org/2020/231.

TKN20. Junichi Tomida, Yuto Kawahara, and Ryo Nishimaki. Fast, compact, and expressive attribute-based encryption. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors, *Public-Key Cryptography - PKC 2020 - 23rd IACR International Conference on Practice and Theory of Public-Key Cryptography, Edinburgh, UK, May 4-7, 2020, Proceedings, Part I*, volume 12110 of *Lecture Notes in Computer Science*, pages 3–33. Springer, 2020.

Tsa19. Rotem Tsabary. Fully secure attribute-based encryption for t-CNF from LWE. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology - CRYPTO 2019*, volume 11692 of *Lecture Notes in Computer Science*, pages 62–85. Springer, 2019.

Ver01. Eric R. Verheul. Evidence that XTR is more secure than supersingular elliptic curve cryptosystems. In Birgit Pfitzmann, editor, *Advances in Cryptology - EUROCRYPT 2001, International Conference on the Theory and Application of Cryptographic Techniques, Innsbruck, Austria, May 6-10, 2001, Proceeding*, volume 2045 of *Lecture Notes in Computer Science*, pages 195–210. Springer, 2001.

Wat09. Brent Waters. Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In Shai Halevi, editor, *Advances in Cryptology - CRYPTO 2009*, volume 5677 of *Lecture Notes in Computer Science*, pages 619–636. Springer, 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, *Public Key Cryptography - PKC 2011*, volume 6571 of *Lecture Notes in Computer Science*, pages 53–70. Springer, 2011.

Wat12. Brent Waters. Functional encryption for regular languages. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology - CRYPTO 2012*, volume 7417 of *Lecture Notes in Computer Science*, pages 218–235. Springer, 2012.

Wee14. Hoeteck Wee. Dual system encryption via predicate encodings. In Yehuda Lindell, editor, *Theory of Cryptography Conference - TCC 2014*, volume 8349 of *Lecture Notes in Computer Science*, pages 616–637. Springer, 2014.

WFL19. Zhedong Wang, Xiong Fan, and Feng-Hao Liu. FE for inner products and its application to decentralized ABE. In Dongdai Lin and Kazue Sako, editors, *Public-Key Cryptography - PKC 2019*, volume 11443 of *Lecture Notes in Computer Science*, pages 97–127. Springer, 2019.