Fully Adaptive Decentralized Multi-Authority ABE

Pratish Datta[1][0000−0002−3938−7594], Ilan Komargodski[1,2][0000−0002−1647−2112], and Brent Waters[1,3]

[1] NTT Research, Sunnyvale, CA 94085, USA
pratish.datta@ntt-research.com,
[2] Hebrew University of Jerusalem, 91904 Jerusalem, Israel
ilank@cs.huji.ac.il,
[3] University of Texas at Austin, Austin, TX 78712, USA
bwaters@cs.utexas.edu

Abstract. Decentralized multi-authority attribute-based encryption (MA-ABE) is a distributed generalization of standard (ciphertext-policy) attribute-based encryption where there is no trusted central authority: any party can become an authority and issue private keys, and there is no requirement for any global coordination other than the creation of an initial set of common reference parameters.

We present the first multi-authority attribute-based encryption schemes that are provably fully-adaptively secure. Namely, our construction is secure against an attacker that may corrupt some of the authorities as well as perform key queries adaptively throughout the life-time of the system. Our main construction relies on a prime order bilinear group where the k-linear assumption holds as well as on a random oracle. Along the way, we present a conceptually simpler construction relying on a composite order bilinear group with standard subgroup decision assumptions as well as on a random oracle.

Prior to this work, there was no construction that could resist adaptive corruptions of authorities, no matter the assumptions used. In fact, we point out that even standard complexity leveraging style arguments do not work in the multi-authority setting.

1 Introduction

Attribute-based encryption schemes [41, 22] allow fine-grained access control when accessing encrypted data: Such encryption schemes support decryption keys that allow users that have certain credentials (or attributes) to decrypt certain messages without leaking any additional information. Over the years, the challenge of designing ABE schemes has received tremendous attention resulting in a long sequence of works achieving various trade-offs between expressiveness, efficiency, security, and underlying assumptions.

Multi-Authority Attribute-Based Encryption: In ABE schemes, restricted decryption keys can only be generated and issued by a central authority who possesses the master secret key. Chase [10] introduced the notion of multi-authority

ABE (MA-ABE) which allows multiple parties to play the role of an authority. More precisely, 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.

After few initial attempts [10, 32, 34, 11, 35] that had various limitations, Lewko and Waters [30] 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. Furthermore, utilizing the powerful dual system technique [45], security is proven assuming a composite order bilinear group with “subgroup decision”-style assumptions and in the random oracle model.

Following Lewko and Waters [30] there were several extensions and improvements. Okamoto and Takashima [38] gave a construction over prime order bilinear groups relying on the decision-linear (DLIN) [7] assumption. Rouselakis and Waters [40] and Ambrona and Gay [2] provided efficiency improvements but provide weaker security guarantees and/or used the less standard q-type assumptions and the generic group model (GGM) respectively. Datta et al. [14] gave the first Learning With Errors (LWE)-based construction supporting a non-trivial class of access policies. All of the above are in the random oracle model. Very recently, Waters, Wee, and Wu [48] gave a construction (for the same class of policies as [14]) whose security can be based in the plain model without random oracles, relying on the recently-introduced evasive LWE assumption [49,43] which is a very strong knowledge type assumption.

Security: The natural MA-ABE security definition requires the usual collusion resistance against unauthorized users with the important difference that now some of the attribute authorities may be corrupted and therefore may collude with the adversarial users. While some constructions support adaptive key queries, *there is no known construction, under any assumption, which supports* **fully adaptive corruption of authorities**. Given the distributed nature of MA-ABE it seems unsatisfying to assume that an attacker commits on a corrupted set of authorities at the beginning of the security game, even before seeing any secret key. Indeed, in reality we do not even expect all attribute authorities to join the system at the same time. Therefore, we argue that the “static corruptions” model that previous works have considered does not capture realistic attack scenarios, and we therefore ask whether it is possible to improve it by supporting adaptive corruption of authorities.

We emphasize that getting fully adaptive security is a well-known gap in existing constructions. Even though the authors of [30] were well versed in sophisticated dual system technique, they (and all followup attempts) got fundamentally stuck in solving this obstacle. More broadly, getting adaptive security is a fundamental area of research in the cryptographic community with many successes over the years (e.g., [47,3,21,33,26]). Still, this natural question in the MA-ABE domain remained untouched.

Interestingly, this is one of the rare cases where generic complexity leveraging/guessing style arguments fail (even if we are fine with a sub-exponential security loss). Indeed, applying these arguments in our setting results in an exponential loss proportional to the maximum number of authorities per ciphertext. Thus, there needs to be a pre-determined maximum number of authorities per ciphertext limit and then the security parameter needs to be chosen appropriately. Our goal, of course, is to devise a truly decentralized scheme where any party could join as an authority at any point in time and there is no limit to the number of authorities.

1.1 Our Results

We construct the first truly decentralized MA-ABE schemes which is provably secure even when fully adaptive corruption of authorities are allowed, in addition to fully adaptive key queries. Our schemes are based on bilinear groups with standard polynomial hardness assumptions and in the random oracle model. We emphasize that our constructions are the first provably secure schemes against fully adaptive corruptions of authorities *under any assumption.*

We first give a construction based on bilinear groups of composite order with (by now) standard subgroup-decision assumptions, and then give a construction in prime order bilinear groups where the k-Linear (k-Lin) [24,42] or more generally the matrix Diffie-Hellman (MDDH) [17] holds.

Theorem 1.1 (Informal; see Section 4): *Assume a composite order bilinear group where "standard" subgroup-decision assumptions hold. Then, there is a fully-adaptive* MA-ABE *scheme in the random oracle model.*

The assumptions that we use in the above theorem have been used multiple times in the past and they were shown to hold in the generic bilinear group model [29,31,30]. However, we still point out that composite order-based constructions have few drawbacks compared to the more standard prime order setting. First, in prime order groups, we can obtain security under more standard assumptions such as k-LIN or bilinear Diffie-Hellman (BDH) [8] assumption. Second, in prime order groups, we can achieve much more efficient systems for the same security levels [19,23,39]. This is because in composite order groups, security typically relies on the hardness of factoring the group order. In turn, this requires the use of large group orders, which results in considerably slower group and pairing operations.

To this end, starting with Freeman [19] and Lewko [27], multiple frameworks and tools have been developed to translate existing composite order group constructions into prime order analogues (see, for example, [36,37,25,4,12,20,1,13]). We use a recent set of tools due to Chen, Gong, Kowalczyk, and Wee [13] (building on [12, 20]) and manage to obtain a construction in (asymmetric) bilinear groups of prime order whose security is based on the more standard k-Lin or MDDH assumption.[4]

Theorem 1.2 (Informal; see Section 5): *Assume a prime order bilinear group where the k-Lin or MDDH assumption holds. Then, there is a fully-adaptive $\mathsf{MA\text{-}ABE}$ scheme in the random oracle model.*

The state of the art $\mathsf{MA\text{-}ABE}$ constructions are compared in Table 1.

Table 1. State of the Art in Decentralized $\mathsf{MA\text{-}ABE}$

Scheme	Access policy	Assumption	Security	Bounded policy size
[2]	NC^1	GGM	adaptive	no
[2]	NC^1	SXDH	selective	no
[30]	NC^1	subgroup decision	adaptive	no
[38]	NC^1	DLin	adaptive	no
[40]	NC^1	q-type	static	no
[14]	DNF	LWE	static	yes
[15]	NC^1	$\mathsf{C/D\text{-}BDH}$	static	yes
[48]	DNF	evasive LWE	static	yes
This Work	NC^1	subgroup decision	full	no
This Work	NC^1	k-Lin or MDDH	full	no

In this table, static security requires all of the ciphertexts, secret keys, and corruption queries to be issued by the adversary before the public key of any attribute authority is published, selective security requires the ciphertext and corruption queries to be made upfront while the key queries can be made adaptively, adaptive security requires corruption queries to be issued ahead of time, but all other queries (secret keys and ciphertexts) can be made adaptively, and full security enables all queries, including corruption queries, to be made adaptively. Schemes having a restriction that the maximal size of policies has to be declared during system setup are said to have bounded policy size. All of the works are in the random oracle model except [48]. Lastly, we mention that this table only lists truly decentralized schemes with no trusted centralized authority.

Technical highlight: As all previous group-based decentralized $\mathsf{MA\text{-}ABE}$ systems secure against adaptive key queries in the standard model [30, 38], we also

[4] Our construction is secure based on any choice of k. For instance, setting $k = 1$ we get security under the Symmetric External Diffie-Hellman Assumption (SXDH), and setting $k = 2$ corresponds to security under the DLIN assumption.

use the dual-systems methodology. However, as we explain below, the existing techniques in this space cannot be used to prove fully adaptive security, that is, security against both adaptive key queries and adaptive corruption of attribute authorities. As our main conceptual contribution, we introduce a new technique within this space that allows us to bleed information from one sub-group to another in an unnoticeable way. We call this technique *dual systems with dual sub-systems* and it allows us to undetectably move information between different sub-groups across ciphertexts and key components via a secondary dual sub-system. We believe that this conceptual contribution is of independent interest. See Section 2 for details.

2 Technical Overview

This section starts by providing an overview of the notion of MA-ABE schemes and our fully adaptive security definition, followed by an exposition of previous works and why they failed to achieve the fully adaptive security. We then continue with explaining our main new ideas, followed by an overview of the final scheme and its security proof. We decided to provide an extensive and detailed technical overview in order to help in understanding the challenges stemming from the fully adaptive security model and our approach for dealing with them. A reader interested in our constructions can directly refer to Section 2.4.1.

2.1 Background on MA-ABE

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 note that this restriction can be relaxed to support an a priori bounded number of attributes per authority [30].) We denote the authority universe by $\mathcal{AU}$.

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) [6, 30]. A decentralized MA-ABE scheme consists of 5 procedures $\mathsf{GlobalSetup}, \mathsf{AuthSetup}, \mathsf{KeyGen}, \mathsf{Enc}$, and Dec. The GlobalSetup procedure gets as input the security parameter (in unary encoding) and outputs global public 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 master secret key pair, $(\mathsf{PK}_u, \mathsf{MSK}_u)$. An authority holding the master secret key MSK_u can then 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,[5] 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.

2.2 Fully Adaptive Security

Just like standard ABE, the security of an MA-ABE scheme demands collusion resistance, that is, no group of colluding users, none of whom is individually authorized to decrypt a ciphertext, should be able to decrypt the same when they pull their secret key components together. However, in case of MA-ABE, it is further required that collusion resistance should hold even if some of the 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. We are aiming for **fully adaptive security** which is roughly defined by the following game:

- **Global Setup:** The challenger runs GlobalSetup to generate global public parameters.
- **Query Phase I:** The attacker is allowed to adaptively make a polynomial number of queries of the following form:
  1. *Authority Setup Query* : the challenger runs AuthSetup to create a public/master key pair for an authority specified by the adversary.
  2. *Secret Key Query* : the challenger runs KeyGen to create a secret key for a given attribute.
  3. *Authority Master Key Query* : the challenger provides the attacker the master secret key corresponding to some authority of the adversary's choice.
- **Challenge Phase:** The adversary submits two messages $\mathsf{msg}_0, \mathsf{msg}_1$, and an access structure along with a set of public keys of authorities involved in the access structure. The authority public keys supplied by the attacker can potentially be malformed, i.e., can fall outside the range of AuthSetup. It

[5] The very recent construction of Waters, Wee, and Wu [48] is in the plain model, however, as mentioned, it is based on a newly introduced and less standard assumption and achieves the rather weak "static" security definition.

gets back from the challenger an encryption of one of the messages (chosen at random) with respect to the access structure. It is crucial that the adversary does not hold enough secret keys/authority master keys to decrypt a message that is encrypted with respect to the access structure.
- **Query Phase 2:** Same as in Query Phase 1 (while making sure that the constraint from the challenge phase is not violated).
- **Guess:** The attacker submits a guess for which message underlies the challenge ciphertext.

All previous MA-ABE schemes consider a much weaker definition where the adversary must commit during the Global Setup phase on the set of authorities in the system as well as on the subset of corrupted authorities. Already at that point, the private/public key pairs of all non-corrupt authorities are created by the challenger and the public keys are given to the attacker. (That is, during Query Phase I and II, only queries of form 2 (secret key query) are allowed.) Our fully adaptive definition is much more realistic given the distributed nature of MA-ABE.

2.3 Limitations of Previous Works

As in any ABE scheme, the challenge in MA-ABE is to make it collusion resistant. Usually, ABE schemes achieve collusion resistance by using the system's authority who knows a master secret key to "tie" together 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.

In a multi-authority setting, however, we want to satisfy the simultaneous goals of autonomous key generation and collusion resistance. The requirement of autonomous key generation means that standard techniques for key randomization cannot be applied since there is no one party to compile all the pieces together. Furthermore, in a decentralized MA-ABE system each component may come from a different authority, where such authorities have no coordination and are possibly not even aware of each other. To overcome this, all previous decentralized MA-ABE schemes use the output of a public hash function applied on the user's global identity, GID, as the randomness tying together multiple key components issued to a specific user by different authorities.[6]

To see the challenge let us focus on one particular construction due to Lewko and Waters [30]. Although this is the very first truly decentralized MA-ABE scheme, all relevant follow-up works heavily rely on it and therefore suffer from similar problems. The security proof of the [30] construction uses the dual system technique originally developed by Waters [45]. In a dual system, ciphertexts and keys can take on two forms: normal or semi-functional. Semi-functional ciphertexts and keys are not used in the real system, they are only used in the security proof. A normal key can decrypt normal or semi-functional ciphertexts,

[6] [48] is an exception; see Footnote 5.

and a normal ciphertext can be decrypted by normal or semi-functional keys. However, when a semi-functional key is used to decrypt a semi-functional ciphertext, decryption will fail. Security for dual systems is proved using a sequence of "indistinguishable" games. The first game is the real security game (with normal ciphertext and keys). In the next game, the ciphertext is semi-functional, while all the keys are normal. For an attacker that makes q secret key requests, we define q games, where in the k-th one, the first k keys are semi-functional while the remaining keys are normal. In game q, all the keys and the challenge ciphertext given to the attacker are semi-functional. Hence, none of the given keys are useful for decrypting the challenge ciphertext.

The proof of [30] follows this high level approach, but inherently relies on the fact that the corrupted authorities are specified in advance. There, towards the end of the proof, all keys are semi-functional and the challenge ciphertext is also semi-functional. The goal in the last hybrid is to move to a game where the semi-functional challenge ciphertext is of a random message (rather than the original message). For this to be indistinguishable, they need to "shut off" the rows in the matrix of the access policy corresponding to the corrupted authorities. This is done by using an information theoretic tool of choosing a vector which is orthogonal to those rows in the challenge ciphertext (such a vector must exist since the corrupted set must be unauthorized). Effectively, this allows them to completely ignore the existence of authority master keys corresponding to those rows, while for the other rows the inexistence of a secret key was already taken care of when they moved to a game where all keys are semi-functional.

This approach clearly fails when authorities can be corrupted adaptively. Technically, it is impossible to "shut off" the rows corresponding to the corrupted authorities since the latter may not be even known at the time the challenge ciphertext is created since authorities may be corrupted *after* the challenge ciphertext is created where the challenger should be able to give the adversary the corresponding master key. However, with the (proof) approach of Lewko and Waters [30] this is impossible since the challenger (at that point) does not even have a properly formed master key for the authority.

A Fundamental Limitation?: At this point it is useful to step back and try to discern whether and why handling corrupted authorities was a foundational problem of [30] and has remained open for more than a decade. Lewko and Waters create an intricate dual system encryption proof that uses two semi-functional subspaces. Their techniques go beyond the prior methods of [28, 29] to adapt to the demands of the multi-authority setting. Now the question is the following.

Question: *Is the lack of handling authority corruption mostly an oversight that can be addressed by pushing their techniques a tiny bit further or is there a more fundamental barrier?*

The answer to this question can be distilled by making a quick observation about the Lewko-Waters construction. In their construction all user keys are composed of bilinear group elements. Thus, one can execute a dual system encryption proof by applying subgroup decision or k-linear assumptions (depending on the setting) to change the distribution of such groups over the course of a sequence of games as is typically done.

The authority master secret keys however consist solely of *exponents* over the order of the group. The reason for authority keys being exponents is a consequence of the demands of the multi-authority setting. To bring authority keys into the fold of a dual system encryption proof one would need a plan for changing such keys to some kind of semi-functional form. However, there is no trodden path in the dual system encryption literature for doing this for keys formed solely from exponents. Indeed, none of the hardness assumptions seem to align with this goal at all!

Due to these fundamental barriers, the construction and proof of Lewko and Waters dealt with key queries and corrupted authorities separately. For uncorrupted authorities, the proof handles key generation queries via a dual system encryption. In contrast, corrupted authorities were statically "routed around" in the proof so as to not have important information when needed and thus taken "outside" the dual system encryption proof.

In our work, we will show how to overcome this barrier and bring adaptive corruption of authorities into the fold of a dual system encryption proof. Doing so will require both a novel construction and proof ideas. We shall focus on the composite order construction next as this is where most of the new ideas already come up and it is also much easier to describe. We give an overview of how we port the construction to the prime order setting in Section 2.5.

2.4 Overview of Our Approach and Our (Composite Order) Scheme

Looking into the Lewko-Waters [30] MA-ABE scheme and the security proof more closely, we observe that their authority master keys consist of two exponents, namely $\alpha, y \leftarrow \mathbb{Z}_N$ where $N = p_1p_2p_3$ is the order of the underlying composite order group. At the final step of their security proof where they transition from a correctly formed semi-functional ciphertext for the challenge message to one for a completely random message, they simulated the exponents α and y based on the instance of the underlying hard problem. As such, they could not hope to give out those keys to the adversary during the security game. In other words, they could not support adaptive corruption of authorities.

In order to resolve this problem, ultimately, we want to come up with a construction and a corresponding proof strategy that never needs to simulate the authority master keys based on instances of underlying hard problems. Towards this end, we first observe that it is due to their scheme design that Lewko-Waters [30] needed to simulate the authority master keys. More specifically, in each ciphertext, the payload is masked with the group element $e(g_1, g_1)^s$ in the target group for random $s \leftarrow \mathbb{Z}_N$. Next, the ciphertext provides secret shares of the masking factor s according to the underlying access policy in the exponent of $e(g_1, g_1)$ and they mask them with α for the corresponding authorities also in the exponent of $e(g_1, g_1)$. This is done to ensure that during decryption, only the shares corresponding to the attributes possessed by the decryptor can be recovered by canceling out the α part with a collection of appropriate secret keys for user GID.

Now, at the final hybrid transition of their security proof, they utilized an assumption similar to decisional bilinear Diffie-Hellman (DBDH) where they simulate s as abc, where $a, b, c \leftarrow \mathbb{Z}_N$ are random exponents and unknown to the simulator. Therefore, the simulator has to embed the term ab within α so that it can simulate the ciphertext components containing the shares of s by canceling out ab in the exponent.

In order to do away with α and transition to a construction and proof technique that do not require simulating the authority master keys, we consider a new element h from the p_1 subgroup in the global public parameters. Instead of relying on the entropy derived from the exponents α corresponding to the authorities/attributes a user does not possess, we would like to rely on the entropy obtained from this new component h to hide the payload (recall that h is a part of the global public parameters and is *not associated* with any attribute authority). Simulating h based on the underlying hard problem would not affect the simulator's ability to give out authority master keys. So, our initial idea is to simply mask the payloads with $e(g_1, h)^s$ for $s \leftarrow \mathbb{Z}_N$. We then provide ElGamal encryptions of the secret shares of the masking factor s under the corresponding authority master keys, which now consist only of the exponents y. More precisely, we include $C_{1,x} = g_1^{r_x}, C_{2,x} = g_1^{y_{\rho(x)} r_x} g_1^{\sigma_x}$ for all rows x of the associated LSSS access structure $(\boldsymbol{M}, \rho)$.[7] For the user's secret keys, instead of generating it as $g^\alpha \cdot \mathsf{H}(\mathsf{GID})^y$, as in Lewko-Waters construction, we form the secret keys as $(h \cdot \mathsf{H}(\mathsf{GID}))^y$.

The high level idea of the security proof is then to change h from being an element of the p_1 subgroup to being an element of the p_1p_2 subgroup. Then, the factor masking the message would become $e(g_1, h)^s \cdot e(g_2, h)^s$. At this point, we can leverage the entropy of $s \bmod p_2$ to hide the payload in the final game.

Dual systems with dual sub-systems: Unfortunately, the above scheme does not satisfy correctness. This is because, at the time of decryption, while pairing the ciphertext and key components, some additional terms involving the shares of the masking factor s in the exponent of $e(g_1, \mathsf{H}(\mathsf{GID}))$ would remain. In order to cancel out these terms and ensure correctness, we introduce another parallel sub-system where we provide ElGamal encryptions of shares of $-s$ under corresponding authority master keys and provide elements of the form $\mathsf{H}(\mathsf{GID})^y$ as part of the user's secret keys. At the time of decryption, this part will produce $e(g_1, \mathsf{H}(\mathsf{GID}))^{-s}$ that will cancel $e(g_1, \mathsf{H}(\mathsf{GID}))^s$ from the first sub-system.

Now, observe that if the same authority master keys y are used across both the sub-systems, then a user obtaining $(h \cdot \mathsf{H}(\mathsf{GID}))^y$ and $\mathsf{H}(\mathsf{GID})^y$ as parts of its secret key can easily recover h^y which may hamper security. We therefore use two different exponents for the two sub-systems.

Overall, our scheme consists of two sub-systems which we refer to as the "A" sub-system and the "B" sub-system. Accordingly, the master key of an attribute

[7] The ρ function maps between rows of the policy matrix $\boldsymbol{M}$ and the index of the associated authorities/attributes.

authority consists of two random exponents $y_A, y_B \leftarrow \mathbb{Z}_N$. The first sub-system deals with encoding the payload and the shares of the masking factor s, whereas the second sub-system works as a shadow system to cancel out some extra terms during decryption to ensure correctness.

Our security proof proceeds as follows. The first step of our proof is to make a ciphertext semi-functional over the p_3 subgroup. The argument relies on two key facts. (1) Any subset of authorities the attacker compromises will not satisfy the access structure. Thus, the corrupted authorities alone are not enough to (information theoretically) determine if the challenge ciphertext is semi-functional. (2) The keys given out by uncorrupted authorities will not have any component in the order p_3 subgroup, thus they will not help out such an attacker (at this step). Put together, this gives a method to leverage the information theoretic steps in order to handle adaptive corruption of authorities. Our approach uses both computational and information theoretic arguments to step between different hybrid experiments. A critical feature of our security proof is that any step that relies on the attacker's keys not satisfying the access structure will be an information theoretic argument, thereby sidestepping issues related to guessing which authorities are corrupted. (There will of course be multiple computational arguments between and setting up the information theoretic ones.) A similar high-level approach of using information regarding what the adversary corrupts only in information theoretic arguments was used in few previous dual system proofs (e.g., [45, 29, 28]), but here we are able to implement the technique in the (more challenging) distributed setting and enfolding corrupted authorities.

Our approach allows us to establish both semi-functional keys and ciphertexts in a given subspace of the cryptosystem. However, it comes with a big caveat. While the semi-functional argument is established in the p_3 subgroup we had to keep it separate from the ciphertext component blinding the message which lives solely in the p_1 subgroup. At this stage it is therefore unclear that all the work we did will even hide the message at all. Therefore, the next portion of our proof needs to "bleed" the semi-functional portions of the ciphertext into the portions blinding the message. Here again our two sub-system construction crucially comes into play. We will take turns by first bleeding over into one and then into the other.

We call this novel technique as a *dual system with dual sub-systems*. This technique utilizes the semi-functionality within one sub-system to introduce semi-functionality within the other. Then, this will allow us to transform the challenge ciphertext and keys in such a way that the p_2 segment of the special group element h remains information-theoretically hidden to the adversary and so its entropy can then be amplified using a suitable randomness extractor to hide the encrypted message completely.

As we mentioned above, we set the user secret key components for the two sub-system asymmetrically, namely, we multiply the special group element h within the user secret key components that correspond to the first sub-system. But, we do not multiply it within those corresponding to the second sub-system. We crucially leverage this asymmetry in the security proof as follows. We first bleed the semi-functional portions within the p_3 subgroup of the second subsystem into the p_2 subgroup of the same to make the p_2 components semi-functional. After that, we utilize this semi-functionality of the second sub-system to switch the special group element h from being embedded within the user secret key components of the first sub-system to those corresponding to the second sub-system. Once we are done with this step, we then bleed the semi-functional portions within the p_3 subgroup of the first sub-system into the p_2 portions of the same and make the p_2 portions of this sub-system semi-functional. This strategy is crucial since it is not clear how to leverage the dual-system methodology to inject semi-functionality into the p_2 portions of the first sub-system if the group element h is not moved away from this sub-system. At this point, the p_2 segment of the ciphertext component blinding the message becomes completely independent of the p_2 segments of all the other ciphertext and key components. Therefore, we can utilize its entropy to blind the message information-theoretically. For a more detailed overview of our hybrid proof strategy, please refer to Section 2.4.2 below.

We once again emphasize that all applications of the dual system methodology so far only dealt with a single system. The two sub-system design is completely new to this work. Also, as we argued above, full security of MA-ABE seems out of reach using standard previously used dual system techniques (since it is not clear how to bleed the semi-functional portions of the ciphertext components into those blinding the message and make the user keys independent of the special group element h within a single system). As is evident from our work, our new technique is useful and we believe that it will find further uses in other contexts related to adaptive security (for example, constructing adaptively secure functional encryption schemes beyond linear functions under standard group-based assumption).

2.4.1 Our Construction

Recall that our scheme relies on bilinear group $\mathbb{G}$ of composite order N which is a product of three primes, that is, $N = p_1p_2p_3$ with subgroups $\mathbb{G}_{p_1}, \mathbb{G}_{p_2}$, and $\mathbb{G}_{p_3}$. We also make use of a seeded randomness extractor $\mathtt{Ext}$ and let seed be a seed for it. The elements g_1 and h are uniformly random generators of the subgroup $\mathbb{G}_{p_1}$ that along with seed are part of the global parameters GP. H is a global hash function that we model as a random oracle in the security proof.

At a very high level, as is evident from the construction, the encryption algorithm blinds the message msg with the term $\mathtt{Ext}(e(g_1, h)^s, \mathsf{seed})$, where s is a random element in $\mathbb{Z}_N$. The goal in the security proof is to show that given the view of the adversary there is enough entropy left in $e(g_1, h)^s$ so that the message is indeed hidden. There are two secret sharing schemes involved, one of s and the other of $-s$. Let us denote the shares of s with $\sigma_{A,x}$ and the shares of $-s$ with $\sigma_{B,x}$. The decryptor recovers $e(g_1, \mathsf{H}(\mathsf{GID}) \cdot h)^{\sigma_{A,x}}$ and $e(g_1, \mathsf{H}(\mathsf{GID}))^{\sigma_{B,x}}$ by appropriately pairing their keys for attributes and ciphertext components. If the user holds sufficient secret keys to decrypt a ciphertext, the two terms $e(g_1, \mathsf{H}(\mathsf{GID}) \cdot h)^{\sigma_{A,x}}$ and $e(g_1, \mathsf{H}(\mathsf{GID}))^{\sigma_{B,x}}$ can be used to recover $e(g_1, \mathsf{H}(\mathsf{GID}) \cdot$

$h)^s$ and $e(g_1, \mathsf{H}(\mathsf{GID}))^{-s}$ which, if multiplied, give the blinding factor $e(g_1, h)^s$, as necessary.

AuthSetup(GP, $\boldsymbol{u}$): The algorithm chooses random values $y_{A,u}, y_{B,u} \in \mathbb{Z}_N$ and outputs

$$\mathsf{PK}_u = (g_1^{y_{A,u}}, g_1^{y_{B,u}}) \qquad \mathsf{MSK}_u = (y_{A,u}, y_{B,u}).$$

Enc(GP, msg, $(\boldsymbol{M}, \boldsymbol{\rho})$, $\{\mathsf{PK}_{\boldsymbol{u}}\}$): It first chooses a random value $s \leftarrow \mathbb{Z}_N$. It then uses the LSSS access policy[8] $(\boldsymbol{M}, \rho)$ to generate a secret sharing of s where $\sigma_{A,x}$ will be the share for all $x \in [\ell]$, i.e, for all $x \in [\ell]$, let $\sigma_{A,x} = \boldsymbol{M}_x \cdot \boldsymbol{v}_A$, where $\boldsymbol{v}_A \leftarrow \mathbb{Z}_N^d$ is a random vector with s as its first entry and $\boldsymbol{M}_x$ is the x^{th} row of $\boldsymbol{M}$. It additionally creates another secret sharing of $-s$ with respect to the LSSS access policy $(\boldsymbol{M}, \rho)$ where $\sigma_{B,x}$ is the corresponding share for $\rho(x)$ for all $x \in [\ell]$, i.e., for all $x \in [\ell]$, $\sigma_{B,x} = \boldsymbol{M}_x \cdot \boldsymbol{v}_B$, where $\boldsymbol{v}_B \leftarrow \mathbb{Z}_N^d$ is a random vector with $-s$ as its first entry. The procedure generates the ciphertext as follows: For each row $x \in [\ell]$, it chooses random $r_{A,x}, r_{B,x} \leftarrow \mathbb{Z}_N$ and outputs the ciphertext

$$\mathsf{CT} = ((\boldsymbol{M}, \rho), C, \{C_{1,A,x}, C_{2,A,x}, C_{1,B,x}, C_{2,B,x}\}_{x \in [\ell]}),$$

where

$$C = \mathsf{msg} \oplus \mathtt{Ext}(e(g_1, h)^s, \mathsf{seed}), \qquad C_{1,A,x} = g_1^{r_{A,x}} \qquad C_{2,A,x} = g_1^{y_{A,\rho(x)} r_{A,x}} g_1^{\sigma_{A,x}}$$
$$C_{1,B,x} = g_1^{r_{B,x}} \qquad C_{2,B,x} = g_1^{y_{B,\rho(x)} r_{B,x}} g_1^{\sigma_{B,x}}.$$

KeyGen(GP, GID, $\mathsf{MSK}_{\boldsymbol{u}}$): The authority attribute u generates a secret key $\mathsf{SK}_{\mathsf{GID},u}$ for GID as $\mathsf{SK}_{\mathsf{GID},u} = (K_{\mathsf{GID},A,u}, K_{\mathsf{GID},B,u})$, where

$$K_{\mathsf{GID},A,u} = (\mathsf{H}(\mathsf{GID}) \cdot h)^{y_{A,u}} \qquad K_{\mathsf{GID},B,u} = (\mathsf{H}(\mathsf{GID}))^{y_{B,u}}.$$

Dec(GP, CT, GID, $\{\mathsf{SK}_{\mathsf{GID},\boldsymbol{u}}\}$): Decryption takes as input the global parameters GP, the hash function H, a ciphertext CT for an LSSS access structure $(\boldsymbol{M}, \rho)$ with $\boldsymbol{M} \in \mathbb{Z}_N^{\ell \times d}$ and $\rho : [\ell] \to \mathcal{AU}$, the user's global identifier $\mathsf{GID} \in \mathcal{GID}$, and the secret keys $\left\{\mathsf{SK}_{\mathsf{GID},\rho(x)}\right\}_{x \in I}$ 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 coefficients $\{w_x \in \mathbb{Z}_N\}_{x \in I}$ such that $(1, 0, \ldots, 0) = \sum_{x \in I} w_x \cdot \boldsymbol{M}_x$.

For all $x \in I$, the decryption algorithm computes:

$$D_{A,x} = e(C_{2,A,x}, \mathsf{H}(\mathsf{GID}) \cdot h) \cdot e(C_{1,A,x}, K_{\mathsf{GID},A,\rho(x)})^{-1} = e(g_1, \mathsf{H}(\mathsf{GID}) \cdot h)^{\sigma_{A,x}}$$
$$D_{B,x} = e(C_{2,B,x}, \mathsf{H}(\mathsf{GID})) \cdot e(C_{1,B,x}, K_{\mathsf{GID},B,\rho(x)})^{-1} = e(g_1, \mathsf{H}(\mathsf{GID}))^{\sigma_{B,x}}.$$

It computes $D = \prod_{x \in I} (D_{A,x} \cdot D_{B,x})^{w_x} = e(g_1, h)^s$ and outputs $C \oplus \mathtt{Ext}(D, \mathsf{seed}) = \mathsf{msg}$. The proposed scheme is correct by inspection; see Section 4.3 for details.

[8] The access policy $(\boldsymbol{M}, \rho)$ is of the form $\boldsymbol{M} = (M_{x,j})_{\ell \times d} = (\boldsymbol{M}_1, \ldots, \boldsymbol{M}_\ell)^\top \in \mathbb{Z}_N^{\ell \times d}$ and $\rho : [\ell] \to \mathcal{AU}$. The function ρ associates rows of $\boldsymbol{M}$ to authorities. We assume that ρ is an injective function, that is, an authority/attribute is associated with at most one row of $\boldsymbol{M}$. This can be extended to a setting where an attribute appears within an access policy for at most a bounded number of times [46, 30].

2.4.2 Our Security Proof

We now dive into a more detailed look at our security proof. We choose to present an overview of the main steps of our proof interleaved with a running commentary on the intuition behind them. Our goal here is to give a reader both a semi-detailed sense of the proof along side the conceptual ideas driving our approach.

Hyb_0 : We start with the real game.

Hyb_1 : Modify the random oracle to return random elements from $\mathbb{G}_{p_1}$. This modification is clearly indistinguishable under the subgroup decision assumption between $\mathbb{G}_{p_1}$ and $\mathbb{G}$.

After this step all user key material is relegated to the $\mathbb{G}_{p_1}$ subgroup. (Recall h was already in $\mathbb{G}_{p_1}$). One important consequence of this is that for any uncorrupted authority u, both the $y_{A,u}$ and $y_{B,u}$ values modulo p_2 and p_3 are *information theoretically* hidden no matter how many keys the attacker requests from the authority u.

Hyb_2 : Add a $\mathbb{G}_{p_3}$ component to each part of the challenge ciphertext. This transition follows from the subgroup decision assumption between $\mathbb{G}_{p_1}$ and $\mathbb{G}_{p_1 p_3}$.

Hyb_3 : We modify the $\mathbb{G}_{p_3}$ components of $C_{2,A,x}, C_{2,B,x}$ to involve shares of independent secrets instead of correlated ones.

This is an information theoretic step relying on two important facts. (1) That the attacker has no information on $y_{A,u}, y_{B,u} (\text{mod } p_3)$ of any uncorrupted authority u per our step in Hyb_1. The fact that $y_{A,u} \mod p_3$ is hidden (and each authority appears at most once in a ciphertext) means that $C_{2,A,x}$ cannot be distinguished from random in the $\mathbb{G}_{p_3}$ subgroup. Thus, the share is hidden when row x corresponds to an uncorrupted authority u. (2) That the rows of the challenge matrix $(\boldsymbol{M}, \rho)$ associated with the corrupted authorities are unauthorized for decryption. Hence, they are insufficient for learning the value of $s \mod p_3$.

Critically, this step employs an information theoretic argument and therefore there is no issue to how to properly embed a reduction to a computational assumption in the presence of adaptive corruptions. In general, this is a theme in our whole reduction process. Throughout the proof, we separate the computational and information theoretic arguments. The parts of the argument that relate to what the attacker corrupted is only in the information theoretic pieces where adaptivity is not a problem.

After this step the ciphertext begins to have a somewhat semi-functional form in that the $\mathbb{G}_{p_3}$ subgroups are not correlated in the system A and B halves. However, the effect is currently vacuous as none of the keys "look at" the $\mathbb{G}_{p_3}$ subgroup which vanishes upon pairing the keys and ciphertext.

Hyb_4 : Add a $\mathbb{G}_{p_2}$ component to each part of the challenge ciphertext. This transition follows from the subgroup decision assumption between $\mathbb{G}_{p_1}$ and $\mathbb{G}_{p_1 p_2}$.

Hyb_5 : Modify the random oracle to return random elements from $\mathbb{G}_{p_1p_3}$. The proof that this change is indistinguishable actually goes through a sequence of sub-hybrids where we change the oracle queries one by one. Intuitively, changing the random oracle output for a certain GID is akin to making the secret key components for GID to be semi-functional. Thus, the proof will need to leverage the fact that the key components acquired by GID do not satisfy the challenge ciphertext access structure. For each GID the proof will first establish this in the $\mathbb{G}_{p_2}$ subgroup to be "temporarily semi-functional", then use this to move it to the "permanent semi-functional" space in $\mathbb{G}_{p_3}$. Finally, undo the work in the $\mathbb{G}_{p_2}$ space to make it available for moving the next GID over.

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

- First modify the random oracle output $\mathsf{H}(\mathsf{GID}_j)$ to be a random element in $\mathbb{G}_{p_1p_2}$ instead of $\mathbb{G}_{p_1}$. This change is clearly indistinguishable under the subgroup decision assumption between $\mathbb{G}_{p_1}$ and $\mathbb{G}_{p_1p_2}$.
- Modify the $\mathbb{G}_{p_2}$ components of $C_{2,A,x}, C_{2,B,x}$ to involve shares of independent secrets instead of correlated ones. This is again an information theoretic step which uses the fact that the rows of the challenge matrix $(\boldsymbol{M}, \rho)$ associated with the corrupted authorities *in conjunction* with all those rows for which the adversary requests a secret key for GID_j are unauthorized for decryption. The adaptive corruption of the authority as well as the adaptive key requests for GID_j do not cause any problem. We emphasize that since this information theoretic argument is done over the $\mathbb{G}_{p_2}$ subgroup, it does not matter whether the adversary has information about the $\mathbb{G}_{p_3}$ from keys for other global identities. This is the benefit for modifying keys one by one in an isolated subspace.
- Next, modify $\mathsf{H}(\mathsf{GID}_j)$ to be a random element from the whole group $\mathbb{G}$. This transition is indistinguishable under the subgroup decision assumption between $\mathbb{G}_{p_1p_2}$ and $\mathbb{G}$. The work done so far allows us to simulate this transition using the group elements available in the problem instance.
- Modify the $\mathbb{G}_{p_2}$ components of $C_{2,A,x}, C_{2,B,x}$ to again involve shares of correlated secrets instead of independent ones. This is again an information theoretic step similar to the previous one.
- Change the random oracle output $\mathsf{H}(\mathsf{GID}_j)$ to be a random element in $\mathbb{G}_{p_1p_3}$ instead of $\mathbb{G}$. This transition is indistinguishable under the subgroup decision assumption between $\mathbb{G}_{p_1}$ and $\mathbb{G}_{p_1p_2}$.

Note that in the above sequence of sub-hybrids, the $\mathbb{G}_{p_2}$ subgroup is used over and over again to "escort" a value into the $\mathbb{G}_{p_3}$ subgroup. Until this step, this portion of the proof follows closely [30] at a high level although there are differences in the low level details. In particular, unlike [30] which involves a single semi-functional form of the ciphertext, we consider several different semi-functional forms in order to handle a more sophisticated scenario of adaptive authority corruption in addition to the adaptive secret key queries. However, the following steps significantly depart from [30].

Hyb_6 : Sample h from $\mathbb{G}_{p_1p_2}$ instead of $\mathbb{G}_{p_1}$. The indistinguishability follows from the subgroup decision assumption between $\mathbb{G}_{p_1}$ and $\mathbb{G}_{p_1p_2}$ In addition, the challenge ciphertext message is now blinded as

$$C = \mathsf{msg}_b \oplus \mathtt{Ext}(e(g_1,h)^s \cdot \boxed{e(g_2,h)^{s''}}, \mathsf{seed})$$

for random s'' and a generator $g_2 \in \mathbb{G}_{p_2}$. At this point the message is blinded in $\mathbb{G}_{p_2}$ while the semi-functional components are established in the $\mathbb{G}_{p_3}$ subgroups for both keys and ciphertexts. We now need to bleed these over into $\mathbb{G}_{p_2}$ to argue the message is hidden.

Hyb_7 : Make the $C_{1,B,x}, C_{2,B,x}$ parts have shares of an independent random secret in $\mathbb{G}_{p_2}$ rather than one correlated to $C_{1,A,x}, C_{2,A,x}$. This is again an information theoretic step which relies on the fact that the rows of the challenge matrix $\boldsymbol{M}$ labeled by the corrupted authorities are unauthorized for decryption.

We now have that the 'B' side of our cryptosystem is complete for our proof with the secret shared on the 'B' side being uncorrelated in the $\mathbb{G}_{p_2}$ component with both the 'A' share and s'' from C. This step is feasible since the keys in our system are created as $\mathsf{H}(\mathsf{GID})^{y_{B,u}}$. In contrast the '$A$' side has keys created as $(\mathsf{H}(\mathsf{GID}) \cdot h)^{y_{A,u}}$. To decouple the $\mathbb{G}_{p_2}$ component of the 'A' side with s'' we must next effectively move the h value from the 'A' side to 'B' side.

Hyb_{10} :[9] Modify the random oracle output for all the global identifiers GID queried by the adversary as $\mathsf{H}(\mathsf{GID}_j) = P_j \cdot h^{-1}$ for the j^{th} random oracle query where P_j is randomly sampled from $\mathbb{G}_{p_1p_3}$. Once this transition is achieved, we will clearly have $\mathsf{H}(\mathsf{GID}_j) \cdot h = P_j$ for all random oracle queries, i.e., $\mathsf{H}(\mathsf{GID}_j) \cdot h$ involves no $\mathbb{G}_{p_2}$ component. This step is crucial for changing the $\mathbb{G}_{p_2}$ components of $C_{1,A,x}, C_{2,A,x}$ in the subsequent hybrids. This transition is achieved via a sequence of sub-hybrids.

- Modify the j^{th} random oracle query to output random elements from $\mathbb{G}$. The indistinguishability follows from the subgroup decision assumption between $\mathbb{G}_{p_1p_2}$ and $\mathbb{G}$.
- Modify the j^{th} random oracle query to output $R_j \cdot h^{-1}$ where R_j is randomly sampled from $\mathbb{G}$. Observe that since R_j is uniformly sampled from $\mathbb{G}$, this new form of $\mathsf{H}(\mathsf{GID}_j)$ is actually identical to the one in the previous game.
- Modify the j^{th} random oracle query to output $P_j \cdot h^{-1}$ where P_j is randomly sampled from $\mathbb{G}_{p_1p_3}$. The indistinguishability follows from the subgroup decision assumption between $\mathbb{G}_{p_1p_2}$ and $\mathbb{G}$.

[9] In our formal proof presented in the full version [16] this is spread out over Hybrids 8-10. We will condense these for this overview and thus skip two numbers of hybrids. We are however not changing the numbers from those in the formal proof for ease of correlation.

Hyb_{11} : Make the $C_{1,A,x}, C_{2,A,x}$ parts have shares of an independent random secret in $\mathbb{G}_{p_2}$. This is again an information theoretic step similar to the previous one of Hyb_6.

Hyb_{12} : Replace C with a random value unrelated to the message. Due to the work done so far, s'' mod p_2 is information theoretically hidden and so s'' has at least $\log(p_2)$ bits of entropy. The extractor hides the message.

2.5 Porting to Prime Order Groups

As mentioned there have been many works trying to come up with a method to translate existing composite order group constructions into prime order analogues [19, 27, 36, 37, 25, 4, 12, 20, 1, 13]. All of these frameworks are different and have varying levels of simplicity or generality. We use the recent framework of Chen et al. [13] which seems to be the most efficient and (arguably) the simplest to use, and succeed in adapting the construction as well as the proof from the composite order setting to the prime order setting.

This framework, in a high level, shows how to simulate a composite order group and its subgroups using a prime order group while guaranteeing a prime order analogue of various subgroup decision style assumptions. These analogues follow from the standard k-Linear assumption (and more generally, the MDDH assumption [18]). Here, since the translation process is not completely black box and needs to be adapted for the scheme at hand, we need to introduce a few extra technical ideas to handle our specific setting. Specifically, the proof of security of our prime order construction relies not only on subgroup decision style assumptions but also on few information theoretic arguments as well as on the security of a random oracle. Using the framework and making it work on our scheme is fairly technical and systematic; we refer to the technical section for details. Nevertheless, we point out that the high level idea as well as the sequence of hybrids is the same as in the composite order case.

3 Preliminaries

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 $[n] = \{1, \ldots, n\}$.

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 column 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 sub-matrix 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 $(-q/2, q/2]$.

Indistinguishability: Two sequences of random variables $\mathcal{X} = \{\mathcal{X}_\lambda\}_{\lambda\in\mathbb{N}}$ and $\mathcal{Y} = \{\mathcal{Y}_\lambda\}_{\lambda\in\mathbb{N}}$ are *computationally indistinguishable* if for any non-uniform PPT algorithm $\mathcal{A}$ there exists a negligible function $\mathsf{negl}(\cdot)$ such that $|\Pr[\mathcal{A}(1^\lambda, \mathcal{X}_\lambda) = 1] - \Pr[\mathcal{A}(1^\lambda, \mathcal{Y}_\lambda) = 1]| \leq \mathsf{negl}(\lambda)$ for all $\lambda \in \mathbb{N}$.

For two distributions $\mathcal{D}$ and $\mathcal{D}'$ over a discrete domain Ω, the statistical distance between $\mathcal{D}$ and $\mathcal{D}'$ is defined as $\mathsf{SD}(\mathcal{D}, \mathcal{D}') = (1/2) \cdot \sum_{\omega \in \Omega} |\mathcal{D}(\omega) - \mathcal{D}'(\omega)|$. A family of distributions $\mathcal{D} = \{\mathcal{D}_\lambda\}_{\lambda \in \mathbb{N}}$ and $\mathcal{D}' = \{\mathcal{D}'_\lambda\}_{\lambda \in \mathbb{N}}$, parameterized by security parameter λ, are said to be *statistically indistinguishable* if there is a negligible function $\mathsf{negl}(\cdot)$ such that $\mathsf{SD}(\mathcal{D}_\lambda, \mathcal{D}'_\lambda) \leq \mathsf{negl}(\lambda)$ for all $\lambda \in \mathbb{N}$.

3.1 Access Structures and Linear Secret Sharing Schemes

Definition 3.1 (Access Structures, [6, 5]): Let $\mathbb{U}$ be the attribute universe. An access structure on $\mathbb{U}$ is a collection $\mathbb{A} \subseteq 2^{\mathbb{U}} \backslash \emptyset$ 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), [6, 5, 30]): 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 $\rho \colon [\ell] \to \mathbb{U}$, 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 $\{\boldsymbol{\mu}_i = (\boldsymbol{M}\boldsymbol{v}^\top)_i\}_{i \in I}$ 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 does not exists any subset I' of the rows of the matrix $\boldsymbol{M}$ and any coefficients $\{w_i\}_{i \in I'}$ for which the above hold.

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 converts this circuit to 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 by 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 together with their negations. Lewko and Waters [30] 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.

3.2 Strong Randomness Extractors

The *min-entropy* of a random variable X is $\mathbf{H}_\infty(X) = -\log(\max_x \mathsf{Pr}[X = x])$. A *$t$-source* is a random variable X with $\mathbf{H}_\infty(X) \geq t$. The *statistical distance* between two random variables X and Y over a finite domain Ω is $\mathbf{SD}(X, Y) = \frac{1}{2}\sum_{\omega\in\Omega}|\mathsf{Pr}[X = \omega] - \mathsf{Pr}[Y = \omega]|$.

Definition 3.3 (Seeded Randomness Extractor, Definition 6.16 [44]): A function $\mathtt{Ext} : \Omega \times S \to \Gamma$ is a strong (t, ϵ)-*extractor* if for every t-source X on Ω, $\mathbf{SD}((\mathcal{U}_S, \mathtt{Ext}(X, \mathcal{U}_S)), (\mathcal{U}_S, \mathcal{U}_\Gamma)) < \epsilon$.

Theorem 3.1 (Theorem 6.17 [44]): *For every $n, t \in \mathbb{N}$ and $\epsilon > 0$, there exists a strong (t, ϵ)-extractor* $\mathtt{Ext} : \{0, 1\}^n \times \{0, 1\}^d \to \{0, 1\}^m$ *with* $m = t - 2\log(1/\epsilon) - O(1)$ *and* $d = \log(n - t) + 2\log(1/\epsilon) + O(1)$.

3.3 Fully-Adaptive Decentralized MA-ABE for LSSS

A decentralized multi-authority attribute-based encryption ($\mathsf{MA\text{-}ABE}$) system $\mathsf{MA\text{-}ABE} = (\mathsf{GlobalSetup}, \mathsf{AuthSetup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ consists of five procedures whose syntax is given below. The supported access structures that we deal with are ones captured by linear secret sharing schemes (LSSS). We denote by $\mathcal{AU}$ the authority universe and by $\mathcal{GID}$ the universe of global identifiers of the users. We denote by $\mathbb{M}$ the supported message space. Additionally, we assume that each authority controls just one attribute, and hence we would use the terms "authority" and "attribute" interchangeably. This definition naturally generalizes to the situation in which each authority can potentially control an arbitrary (bounded or unbounded) number of attributes (see [30, 40]).

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

- $\mathsf{KeyGen}(\mathsf{GP}, \mathsf{GID}, \mathsf{MSK}_u) \mapsto \mathsf{SK}_{\mathsf{GID},u}$: The key generation algorithm takes as input the global parameters GP, a user's global identifier $\mathsf{GID} \in \mathcal{GID}$, and a master secret key MSK_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 $\mathsf{msg} \in \mathbb{M}$, an LSSS access policy $(\boldsymbol{M}, \rho)$ such that $\boldsymbol{M}$ is a matrix over $\mathbb{Z}_N$ 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 $\{(\mathsf{GID}, u)\}$ 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 $\mathsf{MA\text{-}ABE}$ scheme for LSSS-realizable access structures is said to be *correct* if for every $\lambda \in \mathbb{N}$, every message $\mathsf{msg} \in \mathbb{M}$, 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

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

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

The game consists of the following phases:

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

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

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

Challenge Phase: The attacker submits two messages, $\mathsf{msg}_0, \mathsf{msg}_1 \in \mathbb{M}$ and an LSSS access structure $(\boldsymbol{M}, \rho)$. The attacker also submits the public keys $\{\mathsf{PK}_u\}$ for a subset of attribute authorities appearing in the LSSS access structure $(\boldsymbol{M}, \rho)$. The authority public keys $\{\mathsf{PK}_u\}$ supplied by the attacker can potentially be malformed, i.e., can fall outside the range of $\mathsf{AuthSetup}$. The LSSS access structure $(\boldsymbol{M}, \rho)$ and the authority public keys $\{\mathsf{PK}_u\}$ must satisfy the following constraints.

(a) Let $U_{\mathcal{A}}$ denote the set of attribute authorities for which the attacker supplied the authority public keys $\{\mathsf{PK}_u\}$. Also let $U_{\mathcal{B}}$ denote the set of attribute authorities for which the challenger created the master public key pairs in response to the authority setup query of the attacker so far. Then, it is required that $U_{\mathcal{A}} \cap U_{\mathcal{B}} = \emptyset$.

(b) Let V denote the subset of rows of $\boldsymbol{M}$ labeled by the authorities in $U_{\mathcal{A}}$ plus the authorities for which the attacker made a master key query so far. For each global identifier $\mathsf{GID} \in \mathcal{GID}$, let V_{GID} denote the subset of rows of $\boldsymbol{M}$ labeled by authorities u such that the attacker queried a secret key for the pair (GID, u). For each $\mathsf{GID} \in \mathcal{GID}$, it is required that the rows of $\boldsymbol{M}$ labeled by authorities in $V \cup V_{\mathsf{GID}}$ do not span $(1, 0, \ldots, 0)$.

The challenger flips a random coin $b \leftarrow \{0, 1\}$ and generates a ciphertext CT by running the Enc algorithm that encrypts msg_b under the access structure $(\boldsymbol{M}, \rho)$.

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

Guess: The attacker must submit a guess b' for b. The attacker wins if $b = b'$.

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

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

Definition 3.4 (Fully adaptive security for MA-ABE for LSSS): An MA-ABE scheme for LSSS-realizable access structures is fully adaptively 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 $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{MA\text{-}ABE,fully\text{-}adaptive}}(\lambda) \leq \mathsf{negl}(\lambda)$.

Remark 3.2 (Fully adaptive security of MA-ABE for LSSS in the Random Oracle Model): Similar to [30,40,38], we additionally consider the aforementioned notion of fully adaptive security in the random oracle model. In this context, we assume a global hash function H published as part of the global public parameters and accessible by all the parties in the system, including the attacker. In the security proof, we model H as a random function and allow it to be programmed by the challenger. Therefore, in the fully adaptive security game described above, we further let the adversary adaptively submit H-oracle queries to the challenger, along with the key queries it makes both before and after the challenge ciphertext query.

4 Our Composite Order Group MA-ABE Scheme

In Section 4.1 we recall composite order bilinear groups. In Section 4.2 we give the construction. In Section 4.3 we prove correctness of the construction and we give the security proof in the full version [16] The complexity assumptions on which our security proof relies on are basically different types of subgroup decision assumptions and can also be found in the full version.

4.1 Composite Order Bilinear Groups

Our system relies on composite order bilinear groups, which were first defined in [9]. Particularly, we will rely on a bilinear group $\mathbb{G}$ of composite order N which is a product of three primes, that is, $N = p_1p_2p_3$. Such a group has unique subgroups of order q for all divisor q of N and we will denote such a subgroup as $\mathbb{G}_q$. Also every element $g \in \mathbb{G}$, can be written (uniquely) as the product of an element of $\mathbb{G}_{p_1}$, an element of $\mathbb{G}_{p_2}$, and an element of $\mathbb{G}_{p_3}$. We refer to these elements as the "$\mathbb{G}_{p_1}$ part of g", the "$\mathbb{G}_{p_2}$ part of g", and the "$\mathbb{G}_{p_3}$ part of g", respectively. We shall assume that there is a procedure $\mathcal{G}(1^\lambda)$ that gets as input a security parameter λ and outputs $\mathsf{G} = (N = p_1p_2p_3, \mathbb{G}, \mathbb{G}_T, e)$, where $e \colon \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is a pairing. We assume that the group operations in $\mathbb{G}$ and $\mathbb{G}_T$ as well as the bilinear map e are computable in polynomial time in λ. Further, we assume that e satisfies the following:

1. (Bilinear) $\forall g, h \in \mathbb{G}$, $a, b \in \mathbb{Z}_N$, $e(g^a, h^b) = e(g, h)^{ab}$.
2. (Non-degenerate) $\exists g \in \mathbb{G}$ such that $e(g, g)$ has order N in $\mathbb{G}_T$.

4.2 The Construction

Here, we present our $\mathsf{MA\text{-}ABE}$ for NC^1 construction in composite order bilinear groups. As mentioned, we assume that each authority controls just one attribute, and hence we would use the terms "authority" and "attribute" interchangeably. $\mathsf{\textbf{GlobalSetup}}(\mathbf{1^\lambda})$: The global setup algorithm takes in the security parameter 1^λ encoded in unary. The procedure first chooses primes p_1, p_2, p_3 and let $N = p_1p_2p_3$. Next, it generates a bilinear group $\mathsf{G} = (N, \mathbb{G}, \mathbb{G}_T, e)$ of order N. Let $\mathbb{G}_{p_1}$ be the subgroup of $\mathbb{G}$ of order p_1 and let g_1 and h be uniformly random generators of the subgroup $\mathbb{G}_{p_1}$. We make use of a strong seeded randomness extractor $\mathtt{Ext} : \mathbb{G}_T \times S \to \mathbb{M}$, where $\mathbb{M} \subset \{0,1\}^*$ is the message space and $S \subset \{0,1\}^*$ is the seed space. The algorithm samples a seed $\mathsf{seed} \leftarrow S$. It sets the global parameters $\mathsf{GP} = (\mathsf{G}, g_1, h, \mathsf{seed})$. Furthermore, we make use of a hash function $\mathsf{H} : \{0,1\}^* \to \mathbb{G}$ mapping global identities $\mathsf{GID} \in \mathcal{GID}$ to elements in $\mathbb{G}$.

$\mathsf{\textbf{AuthSetup}}(\mathsf{\textbf{GP}}, \mathsf{\textbf{H}}, \boldsymbol{u})$: Given the global parameters GP, the hash function H, and an authority index $u \in \mathcal{AU}$, the algorithm chooses random values $y_{A,u}, y_{B,u} \in \mathbb{Z}_N$ and outputs

$$\mathsf{PK}_u = (P_{A,u} = g_1^{y_{A,u}}, P_{B,u} = g_1^{y_{B,u}}) \qquad \mathsf{MSK}_u = (y_{A,u}, y_{B,u}).$$

$\mathsf{\textbf{Enc}}(\mathsf{\textbf{GP}}, \mathsf{\textbf{H}}, \mathsf{\textbf{msg}}, (\boldsymbol{M}, \boldsymbol{\rho}), \{\mathsf{\textbf{PK}}_{\boldsymbol{u}}\})$: The encryption algorithm takes as input the global parameters GP, the hash function H, a message $\mathsf{msg} \in \mathbb{M}$ to encrypt, an LSSS access structure $(\boldsymbol{M}, \rho)$, where $\boldsymbol{M} = (M_{x,j})_{\ell \times d} = (\boldsymbol{M}_1, \ldots, \boldsymbol{M}_\ell)^\top \in \mathbb{Z}_N^{\ell \times d}$ and $\rho : [\ell] \to \mathcal{AU}$, 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 $\boldsymbol{M}$.

It first chooses a random value $s \leftarrow \mathbb{Z}_N$. It then uses the LSSS access structure $(\boldsymbol{M}, \rho)$ to generate a secret sharing of s where $\sigma_{A,x}$ will be the share for all $x \in [\ell]$, i.e, for all $x \in [\ell]$, let $\sigma_{A,x} = \boldsymbol{M}_x \cdot \boldsymbol{v}_A$, where $\boldsymbol{v}_A \leftarrow \mathbb{Z}_N^d$ is a random vector with s as its first entry and $\boldsymbol{M}_x$ is the x^{th} row of $\boldsymbol{M}$. It additionally creates another secret sharing of $-s$ with respect to the LSSS access policy $(\boldsymbol{M}, \rho)$ where $\sigma_{B,x}$ is the corresponding share for $\rho(x)$ for all $x \in [\ell]$, i.e., for all $x \in [\ell]$, $\sigma_{B,x} = \boldsymbol{M}_x \cdot \boldsymbol{v}_B$, where $\boldsymbol{v}_B \leftarrow \mathbb{Z}_N^d$ is a random vector with $-s$ as its first entry. The procedure generates the ciphertext as follows: For each row $x \in [\ell]$, it chooses random $r_{A,x}, r_{B,x} \leftarrow \mathbb{Z}_N$ and outputs the ciphertext

$$\mathsf{CT} = ((\boldsymbol{M}, \rho), C, \{C_{1,A,x}, C_{2,A,x}, C_{1,B,x}, C_{2,B,x}\}_{x \in [\ell]}),$$

where

$$C = \mathsf{msg} \oplus \mathtt{Ext}(e(g_1, h)^s, \mathsf{seed}),$$
$$C_{1,A,x} = g_1^{r_{A,x}} \qquad C_{2,A,x} = P_{A,\rho(x)}^{r_{A,x}} g_1^{\sigma_{A,x}} = g_1^{y_{A,\rho(x)} r_{A,x}} g_1^{\sigma_{A,x}}$$
$$C_{1,B,x} = g_1^{r_{B,x}} \qquad C_{2,B,x} = P_{B,\rho(x)}^{r_{B,x}} g_1^{\sigma_{B,x}} = g_1^{y_{B,\rho(x)} r_{B,x}} g_1^{\sigma_{B,x}}.$$

$\mathsf{\textbf{KeyGen}}(\mathsf{\textbf{GP}}, \mathsf{\textbf{H}}, \mathsf{\textbf{GID}}, \mathsf{\textbf{MSK}}_{\boldsymbol{u}})$: The key generation algorithm takes as input the global parameters GP, the hash function H, the user's global identifier $\mathsf{GID} \in$

$\mathcal{GID}$, and the authority's master secret key MSK_u. It generates a secret key $\mathsf{SK}_{\mathsf{GID},u}$ for GID as

$$\mathsf{SK}_{\mathsf{GID},u} = (K_{\mathsf{GID},A,u}, K_{\mathsf{GID},B,u})$$

where $K_{\mathsf{GID},A,u} = (\mathsf{H}(\mathsf{GID}) \cdot h)^{y_{A,u}}$ and $K_{\mathsf{GID},B,u} = (\mathsf{H}(\mathsf{GID}))^{y_{B,u}}$.

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 LSSS access structure $(\boldsymbol{M}, \rho)$ with $\boldsymbol{M} \in \mathbb{Z}_N^{\ell \times d}$ and $\rho : [\ell] \to \mathcal{AU}$ injective, the user's global identifier $\mathsf{GID} \in \mathcal{GID}$, and the secret keys $\{\mathsf{SK}_{\mathsf{GID},u}\}_{u \in \rho(I)}$ 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 $\{w_x \in \mathbb{Z}_N\}_{x \in I}$ such that $(1, 0, \ldots, 0) = \sum_{x \in I} w_x \cdot \boldsymbol{M}_x$.

For all $x \in I$, the decryption algorithm first compute:

$$D_{A,x} = e(C_{2,A,x}, \mathsf{H}(\mathsf{GID}) \cdot h) \cdot e(C_{1,A,x}, K_{\mathsf{GID},A,\rho(x)})^{-1} = e(g_1, \mathsf{H}(\mathsf{GID}) \cdot h)^{\sigma_{A,x}}$$
$$D_{B,x} = e(C_{2,B,x}, \mathsf{H}(\mathsf{GID})) \cdot e(C_{1,B,x}, K_{\mathsf{GID},B,\rho(x)})^{-1} = e(g_1, \mathsf{H}(\mathsf{GID}))^{\sigma_{B,x}}$$

Then compute $D = \prod_{x \in I} (D_{A,x} \cdot D_{B,x})^{w_x} = e(g_1, h)^s$. Finally it outputs $C \oplus \mathtt{Ext}(D, \mathsf{seed}) = \mathsf{msg}$.

Remark 4.1 (On GlobalSetup): Similar to all prior decentralized MA-ABE schemes, our proposed schemes utilize a GlobalSetup algorithm that samples a random string ("setup") with a specific structure (i.e., private coin). This setup string needs to be generated only once, can be reused in different sessions, and the randomness used to generate it is never used subsequently so it can be discarded once the setup string is generated.

Theorem 4.1 (Security of Composite-Order MA-ABE Scheme): *The above* MA-ABE *scheme for* NC^1 *is fully adaptively secure in the random oracle model assuming the various types of sub-group decision assumptions.*

The proof of correctness of the scheme is presented in Section 4.3. The proof of security, i.e., that of Theorem 4.1, is deferred to the full version [16].

4.3 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 $x \in I$, letting $\rho(x)$ be the corresponding authority,

$$\begin{aligned} e(C_{2,A,x}, \mathsf{H}(\mathsf{GID}) \cdot h) &= e(g_1^{y_{A,\rho(x)} r_{A,x}} g_1^{\sigma_{A,x}}, \mathsf{H}(\mathsf{GID}) \cdot h) \\ &= e(g_1^{y_{A,\rho(x)} r_{A,x}}, \mathsf{H}(\mathsf{GID}) \cdot h) \cdot e(g_1^{\sigma_{A,x}}, \mathsf{H}(\mathsf{GID}) \cdot h) \\ &= e(g_1, \mathsf{H}(\mathsf{GID}) \cdot h)^{y_{A,\rho(x)} r_{A,x}} \cdot e(g_1, \mathsf{H}(\mathsf{GID}) \cdot h)^{\sigma_{A,x}}. \end{aligned}$$

Also, for each $x \in I$,

$$\begin{aligned} e(C_{1,A,x}, K_{\mathsf{GID},A,\rho(x)}) &= e(g_1^{r_{A,x}}, (\mathsf{H}(\mathsf{GID}) \cdot h)^{y_{A,\rho(x)}}) \\ &= e(g_1, \mathsf{H}(\mathsf{GID}) \cdot h)^{y_{A,\rho(x)} r_{A,x}}. \end{aligned}$$

Hence,

$$\begin{aligned} D_{A,x} &= e(C_{2,A,x}, \mathsf{H}(\mathsf{GID}) \cdot h) \cdot e(C_{1,A,x}, K_{\mathsf{GID},A,\rho(x)})^{-1} \\ &= \frac{e(g_1, \mathsf{H}(\mathsf{GID}) \cdot h)^{y_{A,\rho(x)} r_{A,x}} \cdot e(g_1, \mathsf{H}(\mathsf{GID}) \cdot h)^{\sigma_{A,x}}}{e(g_1, \mathsf{H}(\mathsf{GID}) \cdot h)^{y_{A,\rho(x)} r_{A,x}}} \\ &= e(g_1, \mathsf{H}(\mathsf{GID}) \cdot h)^{\sigma_{A,x}}. \end{aligned}$$

Similarly,

$$\begin{aligned} D_{B,x} &= e(C_{2,B,x}, \mathsf{H}(\mathsf{GID})) \cdot e(C_{1,B,x}, K_{\mathsf{GID},B,\rho(x)})^{-1} \\ &= \frac{e(g_1, \mathsf{H}(\mathsf{GID}))^{y_{B,\rho(x)} r_{B,x}} \cdot e(g_1, \mathsf{H}(\mathsf{GID}))^{\sigma_{B,x}}}{e(g_1, \mathsf{H}(\mathsf{GID}))^{y_{B,\rho(x)} r_{B,x}}} = e(g_1, \mathsf{H}(\mathsf{GID}))^{\sigma_{B,x}}. \end{aligned}$$

We then have

$$\begin{aligned} D &= \prod_{x \in I} (D_{A,x} \cdot D_{B,x})^{w_x} \\ &= \prod_{x \in I} \left(e(g_1, \mathsf{H}(\mathsf{GID}) \cdot h)^{\sigma_{A,x}}\right)^{w_x} \cdot \left(e(g_1, \mathsf{H}(\mathsf{GID}))^{\sigma_{B,x}}\right)^{w_x} \\ &= \prod_{x \in I} e(g_1, \mathsf{H}(\mathsf{GID}) \cdot h)^{w_x \sigma_{A,x}} \cdot e(g_1, \mathsf{H}(\mathsf{GID}))^{w_x \sigma_{B,x}} \\ &= e(g_1, \mathsf{H}(\mathsf{GID}) \cdot h)^{s} \cdot e(g_1, \mathsf{H}(\mathsf{GID}))^{-s} = e(g_1, h)^s, \end{aligned}$$

where the fourth equality follows since $\sum_{x \in I} w_x \cdot \boldsymbol{M}_x = (1, 0, \ldots, 0)$ and $\sigma_{A,x} = \boldsymbol{M}_x \cdot \boldsymbol{v}_A$ and $\sigma_{B,x} = \boldsymbol{M}_x \cdot \boldsymbol{v}_B$. Thus we have

$$\begin{aligned} C \oplus \mathtt{Ext}(D, \mathsf{seed}) &= \mathsf{msg} \oplus \mathtt{Ext}(e(g_1, h)^s, \mathsf{seed}) \oplus \mathtt{Ext}(e(g_1, h)^s, \mathsf{seed}) \\ &= \mathsf{msg}. \end{aligned}$$

5 Our Prime Order Group MA-ABE Scheme

In Section 5.1 we recall prime order bilinear groups and give the associated notations. In Section 5.2 we give the basis structure of the translation framework. Our construction is based on various subspace assumptions derived from the MDDH assumption [13] and can also be found in the full version [16]. In Section 5.3 we give the construction. The correctness and security proofs are deferred to the full version.

5.1 Prime Order Bilinear Groups and Associated Notations

Notations: Let $\boldsymbol{A}$ be a matrix over the ring $\mathbb{Z}_q$. We use $\mathsf{span}(\boldsymbol{A})$ to denote the column span of $\boldsymbol{A}$, and we use $\mathsf{span}^m(\boldsymbol{A})$ to denote matrices of width m where each column lies in $\mathsf{span}(\boldsymbol{A})$; this means $\boldsymbol{M} \leftarrow \mathsf{span}^m(\boldsymbol{A})$ is a random matrix of width m where each column is chose uniformly from $\mathsf{span}(\boldsymbol{A})$. We use $\mathsf{basis}(\boldsymbol{A})$ to denote a basis of $\mathsf{span}(\boldsymbol{A})$, and we use $(\boldsymbol{A}_1 \parallel \boldsymbol{A}_2)$ to denote the column-wise concatenation of matrices $\boldsymbol{A}_1, \boldsymbol{A}_2$. We let $\boldsymbol{I}$ be the identity matrix and $\boldsymbol{0}$ be a zero matrix whose size will be clear from the context.

Fix a security parameter, for any bilinear group parameter $\mathsf{G} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e)$ and any $i = 1, 2, T$ with $g_T = e(g_1, g_2)$, we write $[\![\boldsymbol{M}]\!]_i$ for $g_i^{\boldsymbol{M}}$ where the exponentiation is element-wise. When bracket notation is used, we denote group operations with $\boxplus$, i.e., $[\![\boldsymbol{M}]\!]_i \boxplus [\![\boldsymbol{N}]\!]_i = [\![\boldsymbol{M}+\boldsymbol{N}]\!]_i$ for matrices $\boldsymbol{M}, \boldsymbol{N}$, and $\boxminus$ as their negatives, i.e., $[\![\boldsymbol{M}]\!]_i \boxminus [\![\boldsymbol{N}]\!]_i = [\![\boldsymbol{M}-\boldsymbol{N}]\!]_i$. Also, we define $\boldsymbol{N} \odot [\![\boldsymbol{M}]\!]_i = [\![\boldsymbol{N}\boldsymbol{M}]\!]_i$ and $[\![\boldsymbol{M}]\!]_i \odot \boldsymbol{N} = [\![\boldsymbol{M}\boldsymbol{N}]\!]_i$. We also slightly abuse notations and use the original pairing notation e to denote the pairing between matrices of group elements as well, i.e., we write $e([\![\boldsymbol{M}]\!]_1, [\![\boldsymbol{N}]\!]_2) = [\![\boldsymbol{M}\boldsymbol{N}]\!]_T$.

Prime Order Bilinear Groups: Let $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$ be three multiplicative cyclic groups of prime order $p = p(\lambda)$ where the group operations are efficiently computable in the security parameter λ and there is no isomorphism between $\mathbb{G}_1$ and $\mathbb{G}_2$ that can be computed efficiently in λ. Let g_1, g_2 be generators of $\mathbb{G}_1, \mathbb{G}_2$ respectively and $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ be an efficiently computable pairing function that satisfies the following properties:

– *Bilinearity*: for all $u \in \mathbb{G}_1, v \in \mathbb{G}_2$ and $a, b \in \mathbb{Z}_p$ it is true that $e(u^a, v^b) = e(u, v)^{ab}$.

– *Non-degeneracy*: $e(g_1, g_2) \neq 1_{\mathbb{G}_T}$, 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 an asymmetric bilinear group $\mathsf{G} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e)$.

5.2 Composite to Prime Order Translation Framework

We want to simulate composite order groups whose order is the product of three primes. Fix parameters $\ell_1, \ell_2, \ell_3, \ell_W \geq 1$. Pick random

$$\boldsymbol{A}_1 \leftarrow \mathbb{Z}_p^{\ell \times \ell_1}, \boldsymbol{A}_2 \leftarrow \mathbb{Z}_p^{\ell \times \ell_2}, \boldsymbol{A}_3 \leftarrow \mathbb{Z}_p^{\ell \times \ell_3}$$

where $\ell := \ell_1 + \ell_2 + \ell_3$. Let $(\boldsymbol{A}_1^* \parallel \boldsymbol{A}_2^* \parallel \boldsymbol{A}_3^*)^\top$ denote the inverse of $(\boldsymbol{A}_1 \parallel \boldsymbol{A}_2 \parallel \boldsymbol{A}_3)$, so that $\boldsymbol{A}_i^\top \boldsymbol{A}_i^* = \boldsymbol{I}$ (known as *non-degeneracy*) and $\boldsymbol{A}_i^\top \boldsymbol{A}_j^* = \boldsymbol{0}$ if $i \neq j$ (known as *orthogonality*).

Correspondence: We have the following correspondence with composite order groups:

$$\begin{aligned} g_i &\mapsto [\![\boldsymbol{A}_i]\!]_1, & g_i^s &\mapsto [\![\boldsymbol{A}_i \boldsymbol{s}]\!]_1 \\ w \in \mathbb{Z}_N &\mapsto \boldsymbol{W} \in \mathbb{Z}_p^{\ell \times \ell_W}, & g_i^w &\mapsto [\![\boldsymbol{A}_i^\top \boldsymbol{W}]\!]_1 \end{aligned}$$

The following statistical lemma is analogous to the Chinese Remainder Theorem, which tells us that $w \bmod p_2$ is uniformly random given g_1^w, g_3^w, where $w \leftarrow \mathbb{Z}_N$:

Lemma 5.1 (statistical lemma): *With probability* $1 - 1/p$ *over* $\boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{A}_3, \boldsymbol{A}_1^*, \boldsymbol{A}_2^*, \boldsymbol{A}_3^*$, *the following two distributions are statistically identical.*

$$\{\boldsymbol{A}_1^\top \boldsymbol{W}, \boldsymbol{A}_3^\top \boldsymbol{W}, \boxed{\boldsymbol{W}}\} \quad \text{and} \quad \{\boldsymbol{A}_1^\top \boldsymbol{W}, \boldsymbol{A}_3^\top \boldsymbol{W}, \boxed{\boldsymbol{W} + \boldsymbol{V}^{(2)}}\}$$

where $\boldsymbol{W} \leftarrow \mathbb{Z}_p^{\ell \times \ell_W}$ *and* $\boldsymbol{V}^{(2)} \leftarrow \mathsf{span}^{\ell_W}(\boldsymbol{A}_2^*)$.

5.3 The Construction

Here, we present our MA-ABE for NC^1 construction in prime order bilinear groups. As mentioned, we assume that each authority controls just one attribute, and hence we would use the terms “authority” and “attribute” interchangeably. **GlobalSetup(1^λ)**: The global setup algorithm takes in the security parameter 1^λ encoded in unary. The procedure first chooses a prime p. Next it generates a bilinear group $\mathsf{G} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e)$ of order p. Let g_1, g_2 be the generators of $\mathbb{G}_1, \mathbb{G}_2$ respectively. We make use of a strong seeded randomness extractor $\texttt{Ext} : \mathbb{G}_T \times S \to \mathbb{M}$, where $\mathbb{M} \subset \{0,1\}^*$ is the message space and $S \subset \{0,1\}^*$ is the seed space. The algorithm samples a seed $\mathsf{seed} \leftarrow S$. Next, the algorithm samples $\boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{A}_3 \leftarrow \mathbb{Z}_p^{3k \times k}, \boldsymbol{h} \leftarrow \mathbb{Z}_p^k$. Let $(\boldsymbol{A}_1^* \| \boldsymbol{A}_2^* \| \boldsymbol{A}_3^*) = \big((\boldsymbol{A}_1 \| \boldsymbol{A}_2 \| \boldsymbol{A}_3)^{-1}\big)^\top$ where $\boldsymbol{A}_1^*, \boldsymbol{A}_2^*, \boldsymbol{A}_3^* \leftarrow \mathbb{Z}_p^{3k \times k}$ such that $\boldsymbol{A}_i^\top \boldsymbol{A}_j^* = \boldsymbol{I}$ if $i = j$, and $\boldsymbol{0}$ if $i \neq j$ for all $i, j \in [3]$. It outputs the global parameters as $\mathsf{GP} = (\mathsf{G}, [\![\boldsymbol{A}_1]\!]_1, H = [\![\boldsymbol{A}_1^* \boldsymbol{h}]\!]_2, \mathsf{seed})$.

Furthermore, we assume that all parties has access to the hash function $\mathsf{H} : \{0,1\}^* \to \mathbb{G}_2^{3k}$ mapping global identifiers $\mathsf{GID} \in \mathcal{GID}$ to random vectors in $\mathbb{G}_2^{3k}$, i.e., for all $\mathsf{GID} \in \mathcal{GID}$ we have $\mathsf{H}(\mathsf{GID}) = [\![\boldsymbol{h}_{\mathsf{GID}}]\!]_2$ for some $\boldsymbol{h}_{\mathsf{GID}} \leftarrow \mathbb{Z}_p^{3k}$.
AuthSetup(GP, $\boldsymbol{u}$): Given the global parameters GP and an authority index $u \in \mathcal{AU}$, the algorithm chooses random matrices $\boldsymbol{W}_{A,u}, \boldsymbol{W}_{B,u} \in \mathbb{Z}_p^{3k \times 3k}$ and outputs

$$\begin{aligned} \mathsf{PK}_u &= (P_{A,u} = \boldsymbol{W}_{A,u}^\top \odot [\![\boldsymbol{A}_1]\!]_1, P_{B,u} = \boldsymbol{W}_{B,u}^\top \odot [\![\boldsymbol{A}_1]\!]_1) \\ &= ([\![\boldsymbol{W}_{A,u}^\top \boldsymbol{A}_1]\!]_1, [\![\boldsymbol{W}_{B,u}^\top \boldsymbol{A}_1]\!]_1) \\ \mathsf{MSK}_u &= (\boldsymbol{W}_{A,u}, \boldsymbol{W}_{B,u}). \end{aligned}$$

Enc(GP, msg, ($\boldsymbol{M}, \rho$), {PK$_u$}): The encryption algorithm takes as input the global parameters GP, a message $\mathsf{msg} \in \mathbb{M}$ to encrypt, an LSSS access structure $(\boldsymbol{M}, \rho)$, where $\boldsymbol{M} = (M_{x,j})_{\ell \times d} = (\boldsymbol{M}_1, \ldots, \boldsymbol{M}_\ell)^\top \in \mathbb{Z}_N^{\ell \times d}$ and $\rho : [\ell] \to \mathcal{AU}$, and public keys of the relevant authorities $\{\mathsf{PK}_u\}$. The function ρ associates rows of $\boldsymbol{M}$ (viewed as column vectors) 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 $\boldsymbol{M}$.

It first samples a random vector $\boldsymbol{d} \leftarrow \mathbb{Z}_p^k$ and random matrices $\boldsymbol{U}_A, \boldsymbol{U}_B \leftarrow \mathbb{Z}_p^{3k \times (d-1)}$. The procedure generates the ciphertext as follows: For each row $x \in$

$[\ell]$, it chooses random vectors $\boldsymbol{s}_{A,x}, \boldsymbol{s}_{B,x} \leftarrow \mathbb{Z}_p^k$ and outputs the ciphertext

$$\mathsf{CT} = ((\boldsymbol{M}, \rho), C, \{C_{1,A,x}, C_{2,A,x}, C_{1,B,x}, C_{2,B,x}\}_{x \in [\ell]}),$$

where $C = \mathsf{msg} \oplus \mathtt{Ext}(e([\![\boldsymbol{A}_1 \boldsymbol{d}]\!]_1, H), \mathsf{seed})$, and

$$\begin{aligned}
C_{1,A,x} &= [\![\boldsymbol{A}_1]\!]_1 \odot \boldsymbol{s}_{A,x} = [\![\boldsymbol{A}_1 \boldsymbol{s}_{A,x}]\!]_1 \\
C_{2,A,x} &= ([\![\boldsymbol{A}_1]\!]_1 \odot \boldsymbol{d} \,\|\, [\![\boldsymbol{U}_A]\!]_1) \odot \boldsymbol{M}_x + [\![\boldsymbol{W}_{A,\rho(x)}^\top \boldsymbol{A}_1]\!]_1 \odot \boldsymbol{s}_{A,x} \\
&= \left[\!\!\left[(\boldsymbol{A}_1 \boldsymbol{d} \,\|\, \boldsymbol{U}_A)\, \boldsymbol{M}_x + \boldsymbol{W}_{A,\rho(x)}^\top \boldsymbol{A}_1 \boldsymbol{s}_{A,x}\right]\!\!\right]_1 \\
C_{1,B,x} &= [\![\boldsymbol{A}_1]\!]_1 \odot \boldsymbol{s}_{B,x} = [\![\boldsymbol{A}_1 \boldsymbol{s}_{B,x}]\!]_1 \\
C_{2,B,x} &= ([\![\boldsymbol{A}_1]\!]_1 \odot (-\boldsymbol{d}) \,\|\, [\![\boldsymbol{U}_B]\!]_1) \odot \boldsymbol{M}_x + [\![\boldsymbol{W}_{B,\rho(x)}^\top \boldsymbol{A}_1]\!]_1 \odot \boldsymbol{s}_{B,x} \\
&= \left[\!\!\left[(-\boldsymbol{A}_1 \boldsymbol{d} \,\|\, \boldsymbol{U}_B)\, \boldsymbol{M}_x + \boldsymbol{W}_{B,\rho(x)}^\top \boldsymbol{A}_1 \boldsymbol{s}_{B,x}\right]\!\!\right]_1 .
\end{aligned}$$

KeyGen(GP, GID, MSK$_{\boldsymbol{u}}$): The key generation algorithm takes as input the global parameters GP, the user's global identifier $\mathsf{GID} \in \mathcal{GID}$, and the authority's master secret key MSK_u. It generates a secret key $\mathsf{SK}_{\mathsf{GID},u}$ for GID as

$$\mathsf{SK}_{\mathsf{GID},u} = (K_{\mathsf{GID},A,u}, K_{\mathsf{GID},B,u})$$

where

$$\begin{aligned}
K_{\mathsf{GID},A,u} &= \boldsymbol{W}_{A,u} \odot (\mathsf{H}(\mathsf{GID}) \cdot H) = [\![\boldsymbol{W}_{A,u} \cdot (\boldsymbol{h}_{\mathsf{GID}} + \boldsymbol{A}_1^* \boldsymbol{h})]\!]_2 \\
K_{\mathsf{GID},B,u} &= \boldsymbol{W}_{B,u} \odot \mathsf{H}(\mathsf{GID}) = [\![\boldsymbol{W}_{B,u} \cdot \boldsymbol{h}_{\mathsf{GID}}]\!]_2
\end{aligned}$$

Dec(GP, CT, GID, $\{$SK$_{\mathsf{GID},\boldsymbol{u}}\}$): Decryption takes as input the global parameters GP, a ciphertext CT for an LSSS access structure $(\boldsymbol{M}, \rho)$ with $\boldsymbol{M} \in \mathbb{Z}_N^{\ell \times d}$ and $\rho : [\ell] \to \mathcal{AU}$ injective, the user's global identifier $\mathsf{GID} \in \mathcal{GID}$, and the secret keys $\{\mathsf{SK}_{\mathsf{GID},u}\}_{u \in \rho(I)}$ 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 $\{w_x \in \mathbb{Z}_N\}_{x \in I}$ such that $(1, 0, \ldots, 0) = \sum_{x \in I} w_x \cdot \boldsymbol{M}_x^\top$.

For all $x \in I$, the decryption algorithm first compute:

$$\begin{aligned}
D_{A,x} &= e(C_{2,A,x}, [\![\boldsymbol{h}_{\mathsf{GID}} + \boldsymbol{A}_1^* \boldsymbol{h}]\!]_2) e(C_{1,A,x}, K_{\mathsf{GID},A,\rho(x)})^{-1} \\
&= \left[\!\!\left[((\boldsymbol{A}_1 \boldsymbol{d} \,\|\, \boldsymbol{U}_A)\, \boldsymbol{M}_x)^\top \cdot (\boldsymbol{h}_{\mathsf{GID}} + \boldsymbol{A}_1^* \boldsymbol{h})\right]\!\!\right]_T \\
D_{B,x} &= e(C_{2,B,x}, [\![\boldsymbol{h}_{\mathsf{GID}}]\!]_2) e(C_{1,B,x}, K_{\mathsf{GID},B,\rho(x)})^{-1} \\
&= \left[\!\!\left[((-\boldsymbol{A}_1 \boldsymbol{d} \,\|\, \boldsymbol{U}_B)\, \boldsymbol{M}_x)^\top \cdot \boldsymbol{h}_{\mathsf{GID}}\right]\!\!\right]_T
\end{aligned}$$

Then compute $D = \prod_{x \in I} (D_{A,x} \cdot D_{B,x})^{w_x} = e([\![\boldsymbol{A}_1 \boldsymbol{d}]\!]_1, H)$. Finally it outputs $C \oplus \mathtt{Ext}(D, \mathsf{seed}) = \mathsf{msg}$.

Theorem 5.1 (Security of Prime-Order MA-ABE Scheme): *Assuming the* MDDH *assumption holds, then all PPT adversary has a negligible advantage in breaking the fully adaptive security of the above* MA-ABE *scheme in the random oracle model.*

References

1. Agrawal, S., Chase, M.: A study of pair encodings: Predicate encryption in prime order groups. In: Theory of Cryptography - 13th International Conference, TCC. pp. 259–288 (2016). https://doi.org/10.1007/978-3-662-49099-0_10
2. Ambrona, M., Gay, R.: Multi-authority ABE, revisited. IACR Cryptology ePrint Archive, Report 2021/1381 (2021), https://eprint.iacr.org/2021/1381
3. Ananth, P., Brakerski, Z., Segev, G., Vaikuntanathan, V.: From selective to adaptive security in functional encryption. In: Advances in Cryptology - CRYPTO. pp. 657–677 (2015). https://doi.org/10.1007/978-3-662-48000-7_32
4. Attrapadung, N.: Dual system encryption framework in prime-order groups via computational pair encodings. In: Advances in Cryptology - ASIACRYPT. pp. 591–623 (2016). https://doi.org/10.1007/978-3-662-53890-6_20
5. Beimel, A.: Secure schemes for secret sharing and key distribution. PhD Thesis, Israel Institute of Technology, Technion, Haifa, Israel (1996), https://technion.primo.exlibrisgroup.com/permalink/972TEC_INST/q1jq5o/alma990021768270203971
6. Benaloh, J.C., Leichter, J.: Generalized secret sharing and monotone functions. In: Advances in Cryptology - CRYPTO. pp. 27–35 (1988). https://doi.org/10.1007/0-387-34799-2_3
7. Boneh, D., Boyen, X., Shacham, H.: Short group signatures. In: Advances in Cryptology - CRYPTO. pp. 41–55 (2004). https://doi.org/10.1007/978-3-540-28628-8_3
8. Boneh, D., Franklin, M.K.: Identity-based encryption from the Weil pairing. In: Advances in Cryptology - CRYPTO. pp. 213–229 (2001). https://doi.org/10.1007/3-540-44647-8_13
9. Boneh, D., Goh, E., Nissim, K.: Evaluating 2-DNF formulas on ciphertexts. In: Theory of Cryptography, TCC. pp. 325–341 (2005). https://doi.org/10.1007/978-3-540-30576-7_18
10. Chase, M.: Multi-authority attribute based encryption. In: Theory of Cryptography Conference - TCC. pp. 515–534 (2007). https://doi.org/10.1007/978-3-540-70936-7_28
11. Chase, M., Chow, S.S.M.: Improving privacy and security in multi-authority attribute-based encryption. In: Conference on Computer and Communications Security - CCS. pp. 121–130 (2009). https://doi.org/10.1145/1653662.1653678
12. Chen, J., Gay, R., Wee, H.: Improved dual system ABE in prime-order groups via predicate encodings. In: Advances in Cryptology - EUROCRYPT. pp. 595–624 (2015). https://doi.org/10.1007/978-3-662-46803-6_20
13. Chen, J., Gong, J., Kowalczyk, L., Wee, H.: Unbounded ABE via bilinear entropy expansion, revisited. In: Advances in Cryptology - EUROCRYPT. pp. 503–534 (2018). https://doi.org/10.1007/978-3-319-78381-9_19
14. Datta, P., Komargodski, I., Waters, B.: Decentralized multi-authority ABE for DNFs from LWE. In: EUROCRYPT. pp. 177–209 (2021). https://doi.org/10.1007/978-3-030-77870-5_7
15. Datta, P., Komargodski, I., Waters, B.: Decentralized multi-authority ABE for NC^1 from Computational-BDH. IACR Cryptol. ePrint Arch. p. 1325 (2021), https://eprint.iacr.org/2021/1325
16. Datta, P., Komargodski, I., Waters, B.: Fully adaptive decentralized multi-authority abe. Cryptology ePrint Archive, Paper 2022/1311 (2022), https://eprint.iacr.org/2022/1311

17. Escala, A., Herold, G., Kiltz, E., Ràfols, C., Villar, J.L.: An algebraic framework for diffie-hellman assumptions. In: Advances in Cryptology - CRYPTO. pp. 129–147 (2013). https://doi.org/10.1007/978-3-642-40084-1_8
18. Escala, A., Herold, G., Kiltz, E., Ràfols, C., Villar, J.L.: An algebraic framework for diffie-hellman assumptions. Journal of Cryptology pp. 242–288 (2017). https://doi.org/10.1007/s00145-015-9220-6
19. Freeman, D.M.: Converting pairing-based cryptosystems from composite-order groups to prime-order groups. In: Advances in Cryptology - EUROCRYPT. pp. 44–61 (2010). https://doi.org/10.1007/978-3-642-13190-5_3
20. Gong, J., Dong, X., Chen, J., Cao, Z.: Efficient IBE with tight reduction to standard assumption in the multi-challenge setting. In: Advances in Cryptology - ASIACRYPT. pp. 624–654 (2016). https://doi.org/10.1007/978-3-662-53890-6_21
21. Gong, J., Wee, H.: Adaptively secure ABE for DFA from k-Lin and more. In: Advances in Cryptology - EUROCRYPT. pp. 278–308 (2020). https://doi.org/10.1007/978-3-030-45727-3_10
22. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: Conference on Computer and Communications Security - CCS. pp. 89–98. ACM (2006). https://doi.org/10.1145/1180405.1180418
23. Guillevic, A.: Comparing the pairing efficiency over composite-order and prime-order elliptic curves. In: Applied Cryptography and Network Security - ACNS. pp. 357–372 (2013). https://doi.org/10.1007/978-3-642-38980-1_22
24. Hofheinz, D., Kiltz, E.: Secure hybrid encryption from weakened key encapsulation. In: Advances in Cryptology - CRYPTO. pp. 553–571 (2007). https://doi.org/10.1007/978-3-540-74143-5_31
25. Kowalczyk, L., Lewko, A.B.: Bilinear entropy expansion from the decisional linear assumption. In: Advances in Cryptology - CRYPTO. pp. 524–541 (2015). https://doi.org/10.1007/978-3-662-48000-7_26
26. Kowalczyk, L., Wee, H.: Compact adaptively secure ABE for NC^1 from k-lin. Journal of Cryptology **33**(3), 954–1002 (2020). https://doi.org/10.1007/s00145-019-09335-x
27. Lewko, A.B.: Tools for simulating features of composite order bilinear groups in the prime order setting. In: Advances in Cryptology - EUROCRYPT. pp. 318–335 (2012). https://doi.org/10.1007/978-3-642-29011-4_20
28. Lewko, A.B., Okamoto, T., Sahai, A., Takashima, K., Waters, B.: Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In: Advances in Cryptology - EUROCRYPT. pp. 62–91 (2010). https://doi.org/10.1007/978-3-642-13190-5_4
29. Lewko, A.B., Waters, B.: New techniques for dual system encryption and fully secure HIBE with short ciphertexts. In: Theory of Cryptography Conference - TCC. pp. 455–479 (2010). https://doi.org/10.1007/978-3-642-11799-2_27
30. Lewko, A.B., Waters, B.: Decentralizing attribute-based encryption. In: Advances in Cryptology - EUROCRYPT. pp. 568–588 (2011). https://doi.org/10.1007/978-3-642-20465-4_31
31. Lewko, A.B., Waters, B.: Unbounded HIBE and attribute-based encryption. In: Advances in Cryptology - EUROCRYPT. pp. 547–567 (2011). https://doi.org/10.1007/978-3-642-20465-4_30
32. Lin, H., Cao, Z., Liang, X., Shao, J.: Secure threshold multi authority attribute based encryption without a central authority. In: Progress in Cryptology - INDOCRYPT. pp. 426–436 (2008). https://doi.org/10.1007/978-3-540-89754-5_33

33. Lin, H., Luo, J.: Compact adaptively secure ABE from k-Lin: Beyond NC^1 and towards NL. In: Advances in Cryptology - EUROCRYPT. pp. 247–277 (2020). https://doi.org/10.1007/978-3-030-45727-3_9
34. Müller, S., Katzenbeisser, S., Eckert, C.: Distributed attribute-based encryption. In: International Conference on Information Security and Cryptology - ICISC 2008. pp. 20–36 (2008). https://doi.org/10.1007/978-3-642-00730-9_2
35. Müller, S., Katzenbeisser, S., Eckert, C.: On multi-authority ciphertext-policy attribute-based encryption. Bulletin of the Korean Mathematical Society **46**, 803–819 (07 2009). https://doi.org/10.4134/BKMS.2009.46.4.803
36. Okamoto, T., Takashima, K.: Fully secure functional encryption with general relations from the decisional linear assumption. In: Advances in Cryptology - CRYPTO. pp. 191–208 (2010). https://doi.org/10.1007/978-3-642-14623-7_11
37. Okamoto, T., Takashima, K.: Fully secure unbounded inner-product and attribute-based encryption. In: Advances in Cryptology - ASIACRYPT. pp. 349–366 (2012). https://doi.org/10.1007/978-3-642-34961-4_22
38. Okamoto, T., Takashima, K.: Decentralized attribute-based encryption and signatures. IEICE Transactions on Fundamentals of Electronics, Communications, and Computer Sciences **103-A**(1), 41–73 (2020). https://doi.org/10.1587/transfun.2019CIP0008
39. de la Piedra, A., Venema, M., Alpár, G.: Abe squared: Accurately benchmarking efficiency of attribute-based encryption. IACR Cryptology ePrint Archive, Report 2022/038 (2022), `https://eprint.iacr.org/2022/038`
40. Rouselakis, Y., Waters, B.: Efficient statically-secure large-universe multi-authority attribute-based encryption. In: Financial Cryptography and Data Security - FC. pp. 315–332 (2015). https://doi.org/10.1007/978-3-662-47854-7_19
41. Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: Advances in Cryptology - EUROCRYPT 2005. pp. 457–473. Springer (2005). https://doi.org/10.1007/11426639_27
42. Shacham, H.: A cramer-shoup encryption scheme from the linear assumption and from progressively weaker linear variants. IACR Cryptol. ePrint Arch. **2007**, 74 (2007), `https://eprint.iacr.org/2007/074`
43. Tsabary, R.: Candidate witness encryption from lattice techniques. CRYPTO 2022 (2022). https://doi.org/10.1007/978-3-031-15802-5_19
44. Vadhan, S.P.: Pseudorandomness. Foundations and Trends in Theoretical Computer Science **7**(1-3), 1–336 (2012). https://doi.org/10.1561/0400000010
45. Waters, B.: Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In: Advances in Cryptology - CRYPTO. pp. 619–636 (2009). https://doi.org/10.1007/978-3-642-03356-8_36
46. Waters, B.: Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In: Public Key Cryptography - PKC. pp. 53–70 (2011). https://doi.org/10.1007/978-3-642-19379-8_4
47. Waters, B.: A punctured programming approach to adaptively secure functional encryption. In: Advances in Cryptology - CRYPTO. pp. 678–697 (2015). https://doi.org/10.1007/978-3-662-48000-7_33
48. Waters, B., Wee, H., Wu, D.: Multi-authority ABE from lattices without random oracles. In: TCC. pp. 651–679 (2022). https://doi.org/10.1007/978-3-031-22318-1_23
49. Wee, H.: Optimal broadcast encryption and CP-ABE from evasive lattice assumptions. In: EUROCRYPT. pp. 217–241 (2022). https://doi.org/10.1007/978-3-031-07085-3_8