Revocation Systems with Very Small Private Keys

Amit Sahai[*] sahai@cs.ucla.edu

Brent Waters [†] bwaters@csl.sri.com

Abstract

In this work, we design a new public key broadcast encryption system, and we focus on a critical parameter of device key size: the amount of the cryptographic key material that must be stored securely on the receiving devices. Our new scheme has ciphertext size overhead $O(r)$, where r is the number of revoked users, and the size of public and private keys is only a *constant* number of group elements from an elliptic-curve group of prime order. All previous work, even in the restricted case of systems based on symmetric keys, required at least $\log n$ keys stored on each device. In addition, we show that our techniques can be used to realize Attribute-Based Encryption (ABE) systems with non-monotonic access formulas, where are key storage is significantly more efficient than previous solutions. Our results are in the standard model under a new, but non-interactive, assumption.

1 Introduction

In a broadcast encryption system [14], a broadcaster encrypts a message such that a particular set S of devices can decrypt the message sent over a broadcast channel. Broadcast systems have a wide range of applications including file systems, group communication, DVD content distribution, and satellite subscription services. In many of these applications, the notion of revocation is important: For example, if a DVD-player's key material is leaked on the Internet, one might want to revoke it from decrypting future disks. In another example, consider a group of nodes communicating sensitive control and sensor information over a wireless network; if any of these nodes becomes compromised we'd like to revoke them from all future broadcasts.

Over the years, this problem has received a great deal of attention, and a number of important variants of the problem have been identified. One important restriction is that of *stateless* receivers - where the secret keys stored in the receivers do not need to be updated over time. In such stateless systems there is no need for a user's device to continuously remain on-line to receive key updates. The stateless feature is extremely useful for many applications, such as DVD players which only receive input from the DVD's that are seen by the user (and therefore cannot be relied on to receive all key updates). Another important variant of the broadcast problem is one where a single system can support multiple broadcasters: In the example of a DVD system, we would not want the DVD encryption system designer to be an active participant in the creation of every valid DVD, and also

[*] This research was supported in part by an Alfred P. Sloan Foundation Research Fellowship, an Intel equipment grant, a Cyber-TA Army grant, and NSF ITR/Cybertrust grants 0205594, 0456717 and 0627781.

[†] Supported by NSF CNS-0749931, CNS-0524252, CNS-0716199; the U.S. Army Research Office under the CyberTA Grant No. W911NF-06-1-0316; and the U.S. Department of Homeland Security under Grant Award Number 2006-CS-001-000001.

we would not want to trust a large number of broadcasting entities with the master keys of the system. For such systems, symmetric-key cryptography is not sufficient, and we need *public-key* broadcast encryption schemes. These concerns arise in many applications beyond DVD systems, and therefore it important to address these issues if possible.

Our results. In this work, we design new broadcast encryption schemes, and we focus on the critical parameter of device key size: the amount of the cryptographic key material that must be stored securely on the receiving devices. Keeping the size of private key storage as low as possible is important as cryptographic keys will often be stored in tamper-resistant memory, which is more costly. This can be especially critical in small devices such as sensor nodes, where maintaining low device cost is particularly crucial.

All previous work, even in the restricted case of systems based on symmetric keys, required at least $\log n$ keys stored on each device. In this work, we give the first secure broadcast system in which device keys are only a small *constant* number of group elements (in fact, just 3 group elements) from an elliptic-curve group of prime order. This is typically orders of magnitude smaller than previous schemes for common parameter settings. Furthermore, our scheme is a public-key stateless broadcast encryption scheme[1], and we work with stateless receivers. We achieve this small device key size without compromising on other critical parameters such as ciphertext length – our ciphertexts will consist of just $O(r)$ group elements, where r is the number of revoked users. This is the same behavior as the previously best-known schemes for revocation.

In addition, we show how our techniques can be applied to achieving efficient Attribute-Based Encryption (ABE) [28] schemes with *non-monotonic* access formulas. Ostrovsky, Sahai, and Waters [26] showed a connection between revocation schemes and how to achieving non-monotonic access formulas in ABE; to negate an attribute in an access formula one applies a revocation scheme using the attribute as an identity to be revoked. Ostrovsky, Sahai, and Waters give a particular instance by adapting the revocation scheme of Naor and Pinkas [25] to the ABE scheme of Goyal et. al [18]. The primary drawback of their scheme is that the private key size of their scheme blows up by a *multiplicative* factor of $\log n$, where n is the maximum number of attributes. More precisely, once the DeMorgan's law transformation is made, each negated attribute in the private key will have $O(\log n)$ group elements. By adapting our new revocation techniques to the Goyal et. al ABE scheme each negated attribute will only cost take two group elements. In practice, for many applications the private key storage will decrease by an order of magnitude.

Our Techniques. The primary challenge in constructing broadcast encryption schemes is to achieve full collusion resilience – to make sure that if all the revoked users combine their key material, that they still cannot decrypt ciphertexts. In our approach, we use techniques from bilinear groups to directly achieve this collusion resilience. Our techniques have two major components.

First, we use a "two equation" method for decryption. A ciphertext will be encrypted such that a certain set $S = \{\mathrm{ID}_1, \ldots, \mathrm{ID}_r\}$ will be revoked from decrypting it. Since the ciphertext consists of $O(r)$ group elements, there will be a ciphertext component for each ID_i. Intuitively, when decrypting, a user ID will apply his secret key to each component. If $\mathrm{ID} \neq \mathrm{ID}_i$ he will get two *independent* equations and be able to extract the ith decryption share. However, if $\mathrm{ID} = \mathrm{ID}_i$

[1] And in fact, our scheme is identity-based: Each device's private key can be based on the device's natural "identity," which could be an arbitrary string like a serial number or even an email address. In most previous schemes, every device had to be assigned a specific number between 1 and n.

(*i.e.* he is revoked), then he will only get two *dependent* equations of a two variable formula and thus be unable to extract the decryption share.

Second, we need to make sure that multiple users cannot collude to decrypt the message. For example, if there is a ciphertext that revokes $S = \{\mathrm{ID}_1, \mathrm{ID}_2\}$, these users might try to decrypt by letting user ID_2 get the first share and user ID_1 obtain the second share. To prevent this attack our key shares are randomized or "personalized" to each user to prevent combination of decryption shares.

Our techniques are in contrast to all previous revocation schemes with small key sizes that we are aware of, which used either combinatorial approaches (either with set systems or tree-based approaches) or polynomial interpolation based approaches (see the related work section for more details on previous work). Instead, we devise a new technique for achieving collusion resilience using novel cancellation techniques based on the power of a bilinear map. A bilinear map of the kind we need can be built, for example, from the Weil pairing on elliptic curve groups. Our new collusion-resilience technique allows us to break the bottlenecks that existed in previous systems.

Our system is shown to be secure under a new non-interactive assumption that we call the decisional q-Multi Exponent Bilinear Diffie-Hellman (q-MEBDH) assumption. We show the assumption to hold in the generic bilinear group model in Appendix A[2]. We prove security in the standard model, showing that a ciphertext that revokes up to r users is secure if the decisional r-MEBDH assumption holds.

1.1 Related Work

Fiat and Naor [14] first introduced the problem of broadcast encryption. In their system they proposed a scheme that is secure against a collusion of t users, where the ciphertext size was $O(t \log^2 t \log n)$. This system and other following work [31, 32, 33, 22, 15, 16], used a combinatorial approach. The this type of approach is that there is an inherent tradeoff between the efficiency of the system and the number, t, of colluders that the system is resistant to. An attacker in the system that compromises more than t users can compromise the security of the scheme.

For systems without a bound on the number of revoked users at setup, there have been two general classes of revocation broadcast schemes. The first stateless tree-based revocation schemes were proposed by Naor, Naor and Lopspeich [24] where they introduced the "subset cover" framework. In their framework users were assigned to leaves in a tree and belonged to different subsets. An encryptor encrypts to the minimum number of subsets that covers all the non-revoked users and none of the revoked ones. The primary challenge is to structure the subsets so that they expressive enough to allow for small ciphertext overhead, yet don't impose large private key overhead on the user. The NNL paper proposed two systems with ciphertext sizes of $O(r \lg n)$ and $O(2r)$ and private key sizes of $O(\lg n)$ and $O(\lg^2 n)$ respectively. These methods were subsequently improved upon in future works by Halvey and Shamir [19] and by Goodrich, Sun, and Tamassia [17], where the GST system gives $O(r)$ size ciphertexts and $O(\lg n)$ size private keys. Dodis and Fazio [13] show how to make the the NNL and Halevy and Shamir systems public key by employing hierarchical identity-based encryption methods. It is unknown how to realize the more efficient GST scheme in the public key setting.

The second class of methods is based on polynomial interpolation in the exponents of group

[2] One might wonder if the security proof of our assumption in the generic group model suggests the need for much larger security parameters, thereby negating the efficiency advantages claimed here; indeed we show that this is *not* the case. See Section 4.1 and Appendix A for more details.

elements and was given by Kurosawa and Desmedt [23] and Naor and Pinkas [25]. In these systems the setup algorithm picks a polynomial of degree d, where d is the maximum number of users that can be revoked. Both the public key and ciphertexts are of size d. Yoo et. al. [36] observe that $\lg(n)$ parallel systems can be used to handle n users with $O(r)$ size private keys, $O(n)$ size public keys and $O(r)$ size ciphertexts.

We note that there are a class of *stateful* encryption schemes known as logical-tree-hierarchy schemes independently discovered by Wallner et al. [34] and Wong [35], which are improved in further work [9, 12, 30]. The drawback of stateful schemes is that if a receiver misses an update it won't be able to decrypt future messages (or this must be corrected somehow). Even so, our stateless solution actually provides a more efficient way to revoke users in the stateful setting than previous schemes.

We remark that two equation techniques are somewhat reminiscent of of those used for knowledge extraction in discrete log proof of knowledge settings [29]. In addition, different types of two equation techniques have been applied in ecash applications (see e.g., [8] and the references therein).

Finally, we also note that [7] proposed the first non-trivial *fully collusion resistant* broadcast encryption scheme; broadcasts to a set of uncompromised users remain secure no matter how many other keys the adversary obtained. (In contrast, our approach and those referenced above would lead to very long ciphertexts if the number of revoked users were very large.) Their scheme allows for broadcasts to an arbitrary set of users where the ciphertexts and private key material are both a constant number of group elements, however, the public key material is linear in the number of users in the system and, moreover, the public key must be accessible by any decryptor in the system. This makes their solution unusable for small devices that cannot store the public key. In comparison, our solution is appropriate for applications, like group encryption, where we expect relatively few devices will be compromised and revoked from the encryption and where we need very small storage.

1.2 Organization

The rest of the paper is organized as follows. In Section 2 we provide the relevant definitions for revocation systems and background information on groups with efficiently computable bilinear maps. We then give the construction of our revocation system in Section 3 and prove its security in Section 4. Finally, we show how to realize a non-monotonic Attribute-Based Encryption system with small private key sizes in Section 5.

2 Background

We begin by providing a security definition for a revocation system, in the identity-based framework. We use definitions that are similar, for example, to the definitions for broadcast encryption used by Boneh, Gentry, and Waters [7]; however we adapt our definition to the Identity-Based setting. Later, we state our complexity assumption.

2.1 Identity-Based Revocation Systems

An encryption system is made up of three randomized algorithms: For simplicity of notation, we assume an implicit security parameter of λ.

Setup. An authority will run the setup algorithm. The algorithm outputs a public key PK and master secret key MSK.

***KeyGen*(MSK, ID).** The key generation algorithm takes in the master secret key MSK and an identity, ID. It generates a private key $\mathrm{SK}_{\mathrm{ID}}$ for the identity.

***Encrypt*(S, PK, M).** The encryption algorithm takes as input a revocation set S of identities along with the public key and a message M to encrypt. It outputs a ciphertext CT such that any user with a key for an identity ID $\notin S$ can decrypt.

***Decrypt*(S, CT, ID, $D_{\mathbf{ID}}$)** The decryption algorithm takes as input a ciphertext CT that was generated for the revocation set S, as well as an identity ID and a private key for it. If ID $\notin S$ the algorithm will be able to decrypt and recover the message M encrypted in the ciphertext.

We now define (chosen plaintext) security of an ID-based revocation encryption system against a static adversary. Security is defined using the following "Revocation Game" between an attack algorithm $\mathcal{A}$ and a challenger, for a revocation set S of identities.

> **Setup.** The challenger runs *Setup* to obtain a public key PK and master secret key MSK. It gives $\mathcal{A}$ the public key PK. In addition, it gives $\mathcal{A}$ the decryption keys d_{ID} for all $ID \in S$.
>
> **Challenge.** The attacker gives the challenger two messages M_0, M_1. Next, the challenger picks a random $b \in \{0,1\}$. The challenger runs algorithm *Encrypt* to obtain CT $\overset{\mathrm{R}}{\leftarrow}$ *Encrypt*(S, PK, M_b). It then gives CT to algorithm $\mathcal{A}$.
>
> **Guess.** Algorithm $\mathcal{A}$ outputs its guess $b' \in \{0,1\}$ for b and wins the game if $b = b'$.

Definition 2.1. We say that a revocation system is (chosen-plaintext) secure if, for all revocations sets S of size polynomial in the security parameter, no polynomial-time adversary can win the "Revocation Game" (defined above) with non-negligible advantage over 1/2.

Our attack models the game where all users in the revoked set S get together and collude (this is because the adversary gets all private keys from the revoked set).

Chosen-Ciphertext Security. We will also consider chosen-ciphertext (CCA) security, where the adversary can also issue decryption queries for ciphertexts that it constructs (as long as the challenge ciphertexts are not equal to the challenge ciphertext). The game is identical to the game above, except decryption queries (for arbitrary revocation sets) are allowed. Our main construction will be chosen-plaintext secure; however it can be made CCA-secure using the techniques of Cannetti, Halevi, and Katz [11].

2.2 Bilinear Maps

We briefly review the necessary facts about bilinear maps and bilinear map groups. We use the following standard notation [20, 21, 4]:

1. $\mathbb{G}$ and $\mathbb{G}_T$ are two (multiplicative) cyclic groups of prime order p;
2. g is a generator of $\mathbb{G}$.
3. $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is a bilinear map.

Let $\mathbb{G}$ and $\mathbb{G}_T$ be two groups as above. A bilinear map is a map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ with the following properties:

1. Bilinear: for all $u, v \in \mathbb{G}$ and $a, b \in \mathbb{Z}$, we have $e(u^a, v^b) = e(u, v)^{ab}$.
2. Non-degenerate: $e(g, g) \neq 1$.

We say that $\mathbb{G}$ is a bilinear group if the group action in $\mathbb{G}$ can be computed efficiently and there exists a group $\mathbb{G}_T$ and an efficiently computable bilinear map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ as above. Note that $e(,)$ is symmetric since $e(g^a, g^b) = e(g, g)^{ab} = e(g^b, g^a)$.

2.3 Complexity Assumptions

To prove the security of our system we use a new assumption that we call the q-decisional Multi-Exponent Bilinear Diffie-Hellman assumption. Our assumption falls within a class of assumptions shown to be secure in the generic group model by Boneh, Boyen, and Goh []. While our assumption is non-standard, we emphasize that it is non-interactive and thus falsifiable.

Let $\mathbb{G}$ be a bilinear group of prime order p. The q-MEBDH problem in $\mathbb{G}$ is stated as follows:

A challenger picks a generator $g \in \mathbb{G}$ and random exponents $s, \alpha, a_1, \ldots, a_r$. The attacker is then given $\vec{y}$=

$$\begin{array}{rcccc}
 & & g, g^s, e(g,g)^{\alpha} & & \\
\forall_{1 \le i,j \le q} & g^{a_i} & g^{a_i s} & g^{a_i a_j} & g^{\alpha / a_i^2} \\
\forall_{1 \le i,j,k \le q, i \neq j} & g^{a_i a_j s} & g^{\alpha a_j / a_i^2} & g^{\alpha a_i a_j / a_k^2} & g^{\alpha a_i^2 / a_j^2}
\end{array}$$

it must remain hard to distinguish $e(g, g)^{\alpha \cdot s} \in \mathbb{G}_T$ from a random element in $\mathbb{G}_T$.

An algorithm $\mathcal{B}$ that outputs $z \in \{0, 1\}$ has advantage ϵ in solving decisional q-parallel BDHE in $\mathbb{G}$ if

$$\left| \Pr\left[\mathcal{B}\big(\vec{y}, T = e(g,g)^{\alpha s}\big) = 0\right] - \Pr\left[\mathcal{B}\big(\vec{y}, T = R\big) = 0\right] \right| \geq \epsilon$$

Definition 2.2. We say that the q decisional Multi- Exponent Bilinear Diffie-Hellman assumption holds if no poly-time adversary has non-negligible advantage in winning the game.

Remark. It is tempting to try to simplify our assumption using previous techniques. For example, we might consider letting choosing a single variable a and substituting all a_j with a^j. Unfortunately, this substitution gives rise to an problem that is insecure.

3 Our Revocation System

We now present our revocation system. Our system has the following features: Both public and private keys are of size independent of the number of users (*i.e.* only a constant number of group elements[3]); the ciphertext only contains $O(r)$ group elements, where r is the number of revoked users.

[3] Indeed, since we are using elliptic curves of prime order, these elements can be quite short.

Intuition. Our construction uses a novel application of a secret sharing in the exponent. Suppose an encryption algorithm needs to create an encryption with a revocation set $S = \mathrm{ID}_1, \ldots, \mathrm{ID}_r$ of r identities. The algorithm will create an exponent $s \in \mathbb{Z}_p$ and split it into r random shares $s_1, \ldots, s_r$ such that $\sum s_i = s$. It will then create a ciphertext such that any user key with $\mathrm{ID} = \mathrm{ID}_i$ will not be able to incorporate the $i - th$ share and thus not decrypt the message.

Our approach presents us with two challenges. First, we need to make sure that a user with revoked identity $\mathrm{ID} = \mathrm{ID}_i$ cannot do anything useful with share i. Second, we need to worry about collusion attacks between multiple revoked users. Suppose a user with $\mathrm{ID} = \mathrm{ID}_i$ and a user with $\mathrm{ID} = \mathrm{ID}_j$ collude to attack a ciphertext. The attack we need to worry about is where user j processes ciphertext share i, while user i processes share j, and then they combine their results.

The first problem is addressed by the method of decryption. For each share, the ciphertext will have two components. A user with $\mathrm{ID} \neq \mathrm{ID}_i$ can use these two components to obtain two linearly independent equations (in the exponent) involving the share s_i (and another variable), which he will use to solve for the share s_i. However, if $\mathrm{ID} = \mathrm{ID}_i$ he will get two linearly dependent equations and not be able to solve the system. We remark that these techniques are somewhat reminiscent of of those used for knowledge extraction in discrete log proof of knowledge settings [29]. In addition, different types of two equation techniques have been applied in ecash applications (see e.g., [8] and the references therein).

To address the second challenge, we randomize each user's private key by an exponent t such that in decryption each user recovers shares $t \cdot s_i$ in the exponent. Thus, we disallow useful collusions in a similar manner to some Identity-Based [10, 5] and Attribute-Based [28, 18, 3] encryption systems. Our construction follows.

3.1 Construction

In the description of our construction we will use a bilinear group $\mathbb{G}$ of prime order p. We will assume that identities are taken from the set $\mathbb{Z}_p$; in practice, of course, we can perform a collision resistant hash from identity strings to $\mathbb{Z}_p$. We now give our construction as a set of four algorithms.

Setup The setup algorithm chooses a group $\mathbb{G}$ of prime order p. It then picks random generators $g, h \in \mathbb{G}$ and picks random exponents $\alpha, b \in \mathbb{Z}_p$. The public key is published as:

$$\mathrm{PK} = (g, g^b, g^{b^2}, h^b, e(g,g)^\alpha).$$

The authority keeps α, b as secrets.

Key Gen(MSK, ID) The key generation algorithm first chooses a random $t \in \mathbb{Z}_p$ and publishes the private key as:

$$D_0 = g^\alpha g^{b^2 t}, D_1 = (g^{b \cdot \mathrm{ID}} h)^t, D_2 = g^{-t}.$$

Encrypt(PK, M, S) The encryption algorithm first picks a random $s \in \mathbb{Z}_p$. Then it lets $r = |S|$ and chooses random $s_1, \ldots, s_r$ such that $s = s_1 + \ldots + s_r$. We let ID_i denote the i-th identity in S. It then creates the ciphertext CT as:

$$C' = e(g,g)^{\alpha s} M, C_0 = g^s$$

together with, for each $i = 1, 2, \ldots, r$:

$$\left(C_{i,1} = g^{b \cdot s_i}, C_{i,2} = \left(g^{b^2 \cdot \mathrm{ID}_i} h^b\right)^{s_i}\right)$$

Decrypt(S, CT, ID, $D_{\mathbf{ID}}$) If there exists $\mathrm{ID}' \in S$ such that $\mathrm{ID} = \mathrm{ID}'$ then the algorithm aborts; otherwise, the decryption algorithm computes:

$$\frac{e(C_0, D_0)}{e\left(D_1, \prod_{i=1}^r C_{i,1}^{1/(\mathrm{ID}-\mathrm{ID}_i)}\right) \cdot e\left(D_2, \prod_{i=1}^r C_{i,2}^{1/(\mathrm{ID}-\mathrm{ID}_i)}\right)}$$

which gives us $e(g,g)^{\alpha s}$; this can immediately be used to recover the message M from C'. Note that this computation is only defined if $\forall i \quad \mathrm{ID} \neq \mathrm{ID}_i$.

We can verify the correctness of the decryption computation.

$$\begin{aligned}
& e(C_0, D_0)/\left(e\left(D_1, \prod_{i=1}^r C_{i,1}^{1/(\mathrm{ID}-\mathrm{ID}_i)}\right) \cdot e\left(D_2, \prod_{i=1}^r C_{i,2}^{1/(\mathrm{ID}-\mathrm{ID}_i)}\right)\right) \\
= \quad & e(C_0, D_0)/\left(\prod_{i=1}^r \left(e\left(D_1, C_{i,1}\right) \cdot e\left(D_2, C_{i,2}\right)\right)^{\mathrm{ID}-\mathrm{ID}_i}\right) \\
= \quad & e(g^s, g^\alpha g^{b^2 t})/\left(\prod_{i=1}^r \left(e\left((g^{b\mathrm{ID}} h)^t, g^{bs_i}\right) \cdot e\left(g^{-t}, (g^{b^2 \mathrm{ID}_i} h^b)^{s_i}\right)\right)^{\mathrm{ID}-\mathrm{ID}_i}\right) \\
= \quad & e(g,g)^{s\alpha} e(g,g)^{sb^2 t})/\left(\prod_{i=1}^r e(g,g)^{s_i b^2 t})\right) \\
= \quad & e(g,g)^{s\alpha}
\end{aligned}$$

4 Proof

We now prove the following theorem.

Theorem 4.1. *Suppose the decisional q-MEBDH assumption holds. Then no poly-time adversary can selectively break our system with a ciphertext encrypted to $r^* \leq q$ revoked users.*

Suppose we have an adversary $\mathcal{A}$ with non-negligible advantage $\epsilon = \mathsf{Adv}_{\mathcal{A}}$ in the selective security game against our construction. Moreover, suppose attacks our system with a ciphertext of at most q revoked users. We show how to build a simulator, $\mathcal{B}$, that plays the decisional q-MEBDH problem.

The simulator begins by receiving a q-MEDDH challenge $\vec{X}, T$. The simulator then proceeds in the game as follows.

Init The adversary $\mathcal{A}$ declares a revocation set $S^* = \mathrm{ID}_1, \ldots, \mathrm{ID}_{r^*}$ of size $r^* \leq q$ that he gives to the simulator. (If $r < q$ the simulator will just ignore some of the terms given in $\vec{X}$).

Setup The simulator now creates the public key PK and gives $\mathcal{A}$ the private keys for all identities in S^*. Conceptually, it will set b as $a_1 + a_2 + \cdots a_r$. The simulator first chooses a random $y \in \mathbb{Z}_p$.

The public key PK is published as:

$$\Big(g,\ g^b = \prod_{1 \le i \le r^*} g^{a_i}, \quad g^{b^2} = \prod_{1 \le i,j \le r} (g^{a_i \cdot a_j}), \quad h = \prod_{1 \le i \le r^*} (g^{a_i})^{-\mathrm{ID}_i} g^y, \quad e(g,g)^\alpha\Big)$$

We observe that the public parameters are distributed identically to the real system and that the revocation set S^* is reflected in the simulation's construction of the parameter h.

Now the simulator must construct all private keys in the revocation set S. For each identity ID_i the simulator will choose a random $z_i \in \mathbb{Z}_p$ and will (implicitly) set the randomness t_i of the ith identity as $t_i = -\alpha/a_i^2 + z_i$.

Setting t_i allows us to generate the private key components for two reasons. First, in the D_0 component we need to cancel out the g^α term that we do not know. Since g^{b^2} contains a term of $g^{a_i^2}$ raising it to the $-\alpha/a_i^2$ will cancel this term. Second, we need to make sure that we can still realize the D_2 component. To generate this we will have several terms of the form $g^{\alpha a_j/a_i^2}$, which we have for $i \neq j$. Yet, if $i = j$ this generates a term g^{α/a_i} that we do not have. However, by our setting of the h parameter a term like this will never appear.

The private key for ID_i is generated as follows:

$$D_0 = \left(\prod_{\substack{1 \le j,k \le n \\ \text{s.t. if } j=k \text{ then } j,k \neq i}} (g^{-\alpha a_j a_k / a_i^2}) \right) \prod_{1 \le j,k \le n} (g^{a_j a_k})^{z_i}$$

$$D_1 = \left(\prod_{\substack{1 \le j \le n \\ j \neq i}} (g^{-\alpha \cdot a_j / a_i^2})^{(\mathrm{ID}_i - \mathrm{ID}_j)} (g^{(\mathrm{ID}_i - \mathrm{ID}_j) \cdot a_j})^{z_i} \right) (g^{-\alpha/a_i^2})^y g^{y z_i}$$

$$D_2 = g^{\alpha/a_i^2} g^{-z_i}$$

Remark. Note that in the above construction, for any fixed coefficient μ, by changing $t_i = -\mu\alpha/a_i^2 + z_i$, and appropriately raising the relevant parts of the construction above to a μ factor, one can create $D_0 = g^{\mu\alpha + b^2 t_i}$, while keeping $D_1 = (g^{b\mathrm{ID}_i} h)^{t_i}$, and $D_2 = g^{-t_i}$. This observation is not relevant to this proof, but will be useful in the proof of our related ABE scheme.

Challenge The simulator receives M_0, M_1 and chooses random $\beta \in \{0,1\}$. The simulator then chooses random $s', s'_1, \ldots, s'_{r^*} \in \mathbb{Z}_p$ such that $s' = \sum_i s'_i$. For notational convenience let $u_i = g^{b^2 \mathrm{ID}_i} h^b$, note this is computable from the public parameters, which were already set.

Conceptually, the ciphertext will be encrypted under randomness $\tilde{s} = s + s'$ and be broken into shares $\tilde{s}_i = a_i s/b + s'_i$. Recall, that $b = \sum_j a_j$; therefore, $\sum \tilde{s}_i = \tilde{s}$.

Our methodology is to split s into pieces such that we can simulate all ciphertext components. Conceptually, we will look for a "hole" in each term. We will use the fact that from the simulator's view the function $g^{b\mathrm{ID}_i} h$ has no term of g^{a_i} by cancellation. Therefore, if we raise this to $s \cdot a_i$ the simulator will have all the necessary terms. In this manner we "spread" the different shares of s as $s \cdot a_i/b$, each into its own "slot".

Our proof technique has two important points. First, in simulating the $C_{i,1}$ and $C_{i,2}$ components the b^{-1} term from the shares will cancel out. Second, in generating the $C_{i,2}$ components we will need elements of the form $g^{sa_i a_j}$ that we have for $i \neq j$. Yet, if $i = j$ this creates an element that we do not have. Again, by our setting of h we do not run into this case.

The challenge CT is created as

$$C' = Te(g,g)^{\alpha s'} \cdot M_\beta \quad C_0 = g^s g^{s'} \quad C_{i,1} = g^{sa_i} (\prod_j g^{a_j})^{s'_i} \quad C_{i,2} = \left(\prod_{\substack{1 \leq j \leq r^* \\ i \neq j}} (g^{sa_i a_j})^{\mathrm{ID}_i - \mathrm{ID}_j} \right) (g^{a_i s})^y u_i^{s'_i}$$

The $C_{i,2}$ equation can be understood by recalling that $C_{i,2} = (g^{b\mathrm{ID}_i} h)^{b\tilde{s}_i}$ and then noting that $b\tilde{s}_i = sa_i + s'_i$.

Guess The adversary will eventually output a guess β' of β. The simulator then outputs 0 to guesses that $T = e(g,g)^{\alpha s}$ if $\beta = \beta'$; otherwise, it and outputs 1 to indicate that it believes T is a random group element in $\mathbb{G}_T$.

When T is a tuple the simulator $\mathcal{B}$ gives a perfect simulation so we have that

$$\Pr\left[\mathcal{B}\left(\vec{X}, T = e(g,g)^{\alpha s}\right) = 0\right] = \frac{1}{2} + \mathsf{Adv}_{\mathcal{A}}.$$

When T is a random group element the message M_β is completely hidden from the adversary and we have $\Pr\left[\mathcal{B}\left(\vec{X}, T = R\right) = 0\right] = \frac{1}{2}$. Therefore, $\mathcal{B}$ can play the decisional q-MEBDH game with non-negligible advantage.

4.1 Remark on Security Parameters

Our system is shown to be secure under a new non-interactive assumption. Our proof, in the standard model, shows that a ciphertext that revokes up to r users is secure if the decisional r-MEBDH assumption holds. We remark that generically, an adversary that makes n queries to a group oracle will have advantage $O(n^2 r/p)$ (see Appendix A for a group of prime order p. Equivalent *generic* security to decisional Bilinear Diffie-Hellman can then be realized by increasing the size of p by just an *additive* factor of $\lg(r)$ bits. We recognize, of course, that in general for concrete groups a simpler assumption is desirable, and leave achieving comparable efficiency under simpler assumptions as an important open problem.

5 Attribute-Based Encryption

Our new revocation scheme, as presented in the previous sections, also gives rise to a new efficient Attribute-Based Encryption (ABE) scheme that allows access policies to be expressed in terms of *any* access formula over attributes. Until the recent work of Ostrovsky, Sahai, and Waters [26], all previous ABE schemes were limited to expressing only monotonic access structures. Our new ABE scheme, however, achieves significantly superior parameters in terms of key size. In the random oracle model, our new scheme will have the following key sizes: public parameters will be only $O(1)$ group elements, and private keys for access structures involving t leaf attributes will be of size $O(t)$.

This is a significant improvement over previous work, which needed public parameters consisting of $O(n)$ group elements, and private keys consisting of $O(t\log(n))$ group elements, where n is a bound on the maximum number of attributes that any ciphertext could have. In our scheme, we do not need any such bound.

For brevity, we only describe at a high level what makes our revocation scheme so amenable to incorporation into ABE schemes. The essential property of our revocation scheme is that successful decryption (if a non-revoked user tries to decrypt) allows the user to recover $e(g,g)^{\alpha s}$, where α is a system parameter, while s is a random choice made at the time of encryption. This idea can be applied with α replaced by a linear secret share of α that corresponds to a negated leaf node in an access formula. By the properties of linear secret sharing schemes, and the randomization provided by s, this allows for a secure ABE system to be built using our revocation scheme as a building block.

Taken altogether, our revocation scheme gives a new and much more efficient instiantion of the OSW framework for non-monotonic ABE. We now describe our construction. We refer the reader to [26] for definitions. Our proofs appear in Appendix B.

5.1 Description of ABE construction

We follow the notation of [26] here, and describe our construction in the random oracle model to highlight the most efficient form of our construction.

Setup. The setup algorithm chooses generators g, h and picks random exponents $\alpha', \alpha'', b \in \mathbb{Z}_p$. We define $\alpha = \alpha' \cdot \alpha''$, $g_1 = g^{\alpha'}$ and $g_2 = g^{\alpha''}$.) The public parameters are published as the following, where H is a random oracle that outputs elements of the elliptic curve group:

$$\mathrm{PK} = (g, g^b, g^{b^2}, h^b, e(g,g)^{\alpha}, H(\cdot)).$$

The authority keeps (α', α'', b) as the master key MK.

Encryption (M, γ, PK). To encrypt a message $M \in \mathbb{G}_T$ under a set of d attributes $\gamma \subset \mathbb{Z}_p^*$, choose a random value $s \in \mathbb{Z}_p$, and choose a random set of d values $\{s_x\}_{x\in\gamma}$ such that $s = \sum_{x\in\gamma} s_x$. Output the ciphertext as

$$E = (\gamma, E^{(1)} = Me(g,g)^{\alpha\cdot s}, E^{(2)} = g^s, \{E_x^{(3)} = H(x)^s\}_{x\in\gamma},$$
$$\{E_x^{(4)} = g^{b\cdot s_x}\}_{x\in\gamma}, \{E_x^{(5)} = g^{b^2\cdot s_x x} h^{b\cdot s_x}\}_{x\in\gamma})$$

Key Generation $(\tilde{\mathbb{A}}, \mathrm{MK}, \mathrm{PK})$. This algorithm outputs a key that enables the user to decrypt an encrypted message *only* if the attributes of that ciphertext satisfy the access structure $\tilde{\mathbb{A}}$. We require that the access structure $\tilde{\mathbb{A}}$ is $NM(\mathbb{A})$ for some monotonic access structure $\mathbb{A}$, (see [26] for a definition of the $NM(\cdot)$ operator) over a set $\mathcal{P}$ of attributes, associated with a linear secret-sharing scheme Π. First, we apply the linear secret-sharing mechanism Π to obtain shares $\{\lambda_i\}$ of the secret α'. We denote the party corresponding to the share λ_i as $\breve{x}_i \in \mathcal{P}$, where x_i is the attribute underlying $\breve{x}_i$. Note that $\breve{x}_i$ can be primed (negated) or unprimed (non negated). For each i, we also choose a random value $r_i \in \mathbb{Z}_p$.

The private key D will consist of the following group elements: For every i such that $\breve{x}_i$ is *not* primed (i.e., is a non-negated attribute), we have

$$D_i = (D_i^{(1)} = g_2^{\lambda_i} \cdot H(x_i)^{r_i}, D_i^{(2)} = g^{r_i})$$

For every i such that $\breve{x}_i$ is primed (i.e., is a negated attribute), we have

$$D_i = (D_i^{(3)} = g_2^{\lambda_i} g^{b^2 r_i}, D_i^{(4)} = g^{r_i b x_i} h^{r_i}, D_i^{(5)} = g^{-r_i})$$

The key D consists of D_i for all shares i.

Decryption (E, D). Given a ciphertext E and a decryption key D, the following procedure is executed: (All notation here is taken from the above descriptions of E and D, unless the notation is introduced below.) First, the key holder checks if $\gamma \in \tilde{\mathbb{A}}$ (we assume that this can be checked efficiently). If not, the output is $\perp$. If $\gamma \in \tilde{\mathbb{A}}$, then we recall that $\tilde{\mathbb{A}} = NM(\mathbb{A})$, where $\mathbb{A}$ is an access structure, over a set of parties $\mathcal{P}$, for a linear secret sharing-scheme Π. Denote $\gamma' = N(\gamma) \in \mathbb{A}$, and let $I = \{i : \breve{x}_i \in \gamma'\}$. Since γ' is authorized, an efficient procedure associated with the linear secret-sharing scheme yields a set of coefficients $\Omega = \{\omega_i\}_{i \in I}$ such that $\sum_{i \in I} \omega_i \lambda_i = \alpha$. (Note, however, that these λ_i are not known to the decryption procedure, so neither is α.)

For every positive (non negated) attribute $\breve{x}_i \in \gamma'$ (so $x_i \in \gamma$), the decryption procedure computes the following:

$$\begin{aligned} Z_i &= e\left(D_i^{(1)}, E^{(2)}\right) / e\left(D_i^{(2)}, E_i^{(3)}\right) \\ &= e\left(g_2^{\lambda_i} \cdot H(x_i)^{r_i}, g^s\right) / e\left(g^{r_i}, H(x)^s\right) \\ &= e\left(g, g_2\right)^{s\lambda_i} \end{aligned}$$

For every negated attribute $\breve{x}_i \in \gamma'$ (so $x_i \notin \gamma$), the decryption procedure computes the following, following a simple analogy to the basic revocation scheme:

$$\begin{aligned} Z_i &= \frac{e\left(D_i^{(3)}, E^{(2)}\right)}{e\left(D_i^{(4)}, \prod_{x \in \gamma} \left(E_x^{(4)}\right)^{1/(x_i - x)}\right) \cdot e\left(D_i^{(5)}, \prod_{x \in \gamma} \left(E_x^{(5)}\right)^{1/(x_i - x)}\right)} \\ &= e\left(g, g_2\right)^{s\lambda_i} \end{aligned}$$

Finally, the decryption is obtained by computing

$$\frac{E^{(1)}}{\prod_{i \in I} Z_i^{\omega_i}} = \frac{Me(g,g)^{s\alpha}}{e(g,g_2)^{s\alpha'}} = M$$

Note on Efficiency and Use of Random Oracle Model. We note that encryption requires only a single pairing, which may be pre-computed, regardless of the number of attributes associated with a ciphertext. We also note that decryption requires two or three pairings per share utilized in decryption, depending on whether the share corresponds to a non-negated attribute or a negated attribute, respectively.

We also note that we use a random oracle for description simiplicity and efficiency of the system. We can, alternatively, realize our hash function concretely as in other previous ABE systems [28, 18, 26].

References

[1] H. Anton and C. Rorres. *Elementary Linear Algebra, 9th Edition.* 2005.

[2] A. Beimel. *Secure Schemes for Secret Sharing and Key Distribution.* PhD thesis, Israel Institute of Technology, Technion, Haifa, Israel, 1996.

[3] John Bethencourt, Amit Sahai, and Brent Waters. Ciphertext-policy attribute-based encryption. In *Proceedings of the IEEE Symposium on Security and Privacy (To Appear)*, 2007.

[4] D. Boneh and M. Franklin. Identity Based Encryption from the Weil Pairing. In *Advances in Cryptology – CRYPTO*, volume 2139 of *LNCS*, pages 213–229. Springer, 2001.

[5] Dan Boneh and Xavier Boyen. Efficient Selective-ID Secure Identity Based Encryption Without Random Oracles. In *Advances in Cryptology – Eurocrypt*, volume 3027 of *LNCS*, pages 223–238. Springer, 2004.

[6] Dan Boneh, Xavier Boyen, and Eu-Jin Goh. Hierarchical identity based encryption with constant size ciphertext. In Ronald Cramer, editor, *EUROCRYPT*, volume 3494 of *Lecture Notes in Computer Science*, pages 440–456. Springer, 2005.

[7] Dan Boneh, Craig Gentry, and Brent Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In *CRYPTO*, pages 258–275, 2005.

[8] Jan Camenisch, Susan Hohenberger, and Anna Lysyanskaya. Compact e-cash. In *EUROCRYPT*, pages 302–321, 2005.

[9] R. Canetti, J. Garay, G. Itkis, D. Micciancio, M. Naor, and B. Pinkas. Multicast security: A taxonomy and some efficient constructions. In *Proc. IEEE INFOCOM 1999*, volume 2, pages 708–716. IEEE, 1999.

[10] R. Canetti, S. Halevi, and J. Katz. A Forward-Secure Public-Key Encryption Scheme. In *Advances in Cryptology – Eurocrypt*, volume 2656 of *LNCS*. Springer, 2003.

[11] R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. In *Proc. of Eurocrypt 2004*, volume 3027 of *LNCS*, pages 207–222. Springer-Verlag, 2004.

[12] R. Canetti, T. Malkin, and K. Nissim. Efficient communication-storage tradeoffs for multicast encryption. In *Proc. of Eurocrypt 1999*, pages 459–474. Springer-Verlag, 1999.

[13] Yevgeniy Dodis and Nelly Fazio. Public key broadcast encryption for stateless receivers. In *Digital Rights Management Workshop*, pages 61–80, 2002.

[14] A. Fiat and M. Naor. Broadcast encryption. In *Proc. of Crypto 1993*, volume 773 of *LNCS*, pages 480–491. Springer-Verlag, 1993.

[15] E. Gafni, J. Staddon, and Y.L. Yin. Efficient methods for integrating traceability and broadcast encryption. In *Proc. of Crypto 1999*, volume 1666 of *LNCS*, pages 372–387. Springer-Verlag, 1999.

[16] J. Garay, J. Staddon, and A. Wool. Long-lived broadcast encryption. In *Proc. of Crypto 2000*, volume 1880 of *LNCS*, pages 333–352. Springer-Verlag, 2000.

[17] M.T. Goodrich, J.Z. Sun, and R. Tamassia. Efficient tree-based revocation in groups of low-state devices. In *Proc. of Crypto 2004*, volume 3152 of *LNCS*, pages 511–527. Springer-Verlag, 2004.

[18] V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute Based Encryption for Fine-Grained Access Conrol of Encrypted Data. In *ACM conference on Computer and Communications Security (ACM CCS)*, 2006.

[19] D. Halevy and A. Shamir. The LSD Broadcast Encryption Scheme. In *Advances in Cryptology – CRYPTO*, volume 2442 of *LNCS*, pages 47–60. Springer, 2002.

[20] A. Joux. A one round protocol for tripartite Diffie-Hellman. In *Proc. of ANTS IV*, volume 1838 of *LNCS*, pages 385–94. Springer-Verlag, 2000.

[21] A. Joux and K. Nguyen. Separating decision Diffie-Hellman from Diffie-Hellman in cryptographic groups. *J. of Cryptology*, 16(4):239–247, 2003. Early version in Cryptology ePrint Archive, Report 2001/003.

[22] Ravi Kumar, Sridhar Rajagopalan, and Amit Sahai. Coding constructions for blacklisting problems without computational assumptions. In *CRYPTO*, pages 609–623, 1999.

[23] Kaoru Kurosawa and Yvo Desmedt. Optimum traitor tracing and asymmetric schemes. In *EUROCRYPT*, pages 145–157, 1998.

[24] D. Naor, M. Naor, and J. Lotspiech. Revocation and tracing schemes for stateless receivers. In *Proc. of Crypto 2001*, volume 2139 of *LNCS*, pages 41–62. Springer-Verlag, 2001.

[25] M. Naor and B. Pinkas. Efficient trace and revoke schemes. In *Proc. of Financial cryptography 2000*, volume 1962 of *LNCS*, pages 1–20. Springer-Verlag, 2000.

[26] Rafail Ostrovksy, Amit Sahai, and Brent Waters. Attribute Based Encryption with Non-Monotonic Access Structures. In *ACM conference on Computer and Communications Security (ACM CCS)*, 2007.

[27] V.V. Prasolov. *Problems and Theorems in Linear Algebra.* American Mathematical Society, 1994.

[28] A. Sahai and B. Waters. Fuzzy Identity Based Encryption. In *Advances in Cryptology – Eurocrypt*, volume 3494 of *LNCS*, pages 457–473. Springer, 2005.

[29] Claus-Peter Schnorr. Efficient signature generation by smart cards. *J. Cryptology*, 4(3), 1991.

[30] A.T. Sherman and D.A. McGrew. Key establishment in large dynamic groups using one-way function trees. *IEEE Trans. Softw. Eng.*, 29(5):444–458, 2003.

[31] D.R. Stinson. On some methods for unconditionally secure key distribution and broadcast encryption. *Des. Codes Cryptography*, 12(3):215–243, 1997.

[32] D.R. Stinson and T.V. Trung. Some new results on key distribution patterns and broadcast encryption. *Des. Codes Cryptography*, 14(3):261–279, 1998.

[33] D.R. Stinson and R. Wei. Combinatorial properties and constructions of traceability schemes and frameproof codes. *SIAM J. Discret. Math.*, 11(1):41–53, 1998.

[34] D.M. Wallner, E.J. Harder, and R.C. Agee. Key management for multicast: Issues and architectures. IETF draft wallner-key, 1997.

[35] C.K. Wong, M. Gouda, and S. Lam. Secure group communications using key graphs. In *Proc. of SIGCOMM 1998*, 1998.

[36] E. Yoo, N. Jho, J. Cheon, and M. Kim. Efficient broadcast encryption using multiple interpolation methods. In *Proc. of ICISC 2004*, LNCS. Springer-Verlag, 2004.

A Generic Security of Multi-Exponent BDH

We briefly show that are decisional MEBDH assumption is generically secure. We use the generic proof template of Boneh, Boyen, and Goh [6].

Using the terminology from BBG we need to show that $f = \alpha s$ in independent of the polynomials P and Q. We have that $Q = \{1, \alpha\}$ In addition, we have

$$\begin{aligned} P &= \{1, s, \ \forall_{i,j \in [1,q]} \quad a_i, \ a_i s, \ a_i a_j, \ \alpha/(a_i)^2\} \\ &\cup \ \{\forall_{i,j,k \in [1,q], i \neq j} \quad a_i a_j s, \ \alpha a_j / a_i^2, \alpha a_i a_j / a_k^2, \alpha a_i^2 / a_j^2\} \end{aligned}$$

We first note that this case at first might appear to be outside the BBG framework, since the polynomials are rational function (due to the terms with inverses. However, by a simple renaming of terms we can see this is equivalent to an assumption where we use a generator u and let $g = g^{\prod_{j \in [1,q]} a_j^2}$. Applying this substitution we get a a set of polynomials where maximum degree of any polynomial in the set P is $2q+3$.

We need to also check that f is symbolically independent of the of any two polynomials in P, Q. To realize f from P, Q we would need to have a term of the form αs. We note that no such terms can be realized from the product of two polynomials $p, p' \in P$. If we use the polynomial s as p then no other potential p' has α. If we use $a_i \cdot s$ as p then no other potential p' has α / a_i. Finally, if we use $a_i a_j s$ with $i \neq j$ for p then no other potential p' is of the form $\alpha/(a_i a_j)$ for $i \neq j$. Any dependence on f must have an a term of s in it, but we just eliminated all possibilities.

It follows from the BBG framework that the assumption is then generically secure. In particular, for an attacker that makes at most n queries to the group oracle we have that its advantage is bounded by

$$\frac{(n + 2(q^3 + 4q^2 + 3q) + 2)^2 \cdot (4q + 6)}{2p}$$

In the general case where $n > q^3$ we have that the advantage is $O(n^2 \cdot q/p)$.

B Proof of Security for ABE scheme

We prove that the security of our main construction in the attribute-based selective-set model reduces to the hardness of the q-MEBDH assumption.

Theorem B.1. *If an adversary can break our ABE scheme with advantage ϵ in the attribute-based selective-set model of security, then a simulator can be constructed to play the q-MEBDH game with advantage $\epsilon/2$.*

Proof:

Our proof will follow the outline of, and include much of the text from, the proofs of previous ABE schemes [28, 18, 26], but will incorporate the ideas from our new revocation scheme. We note that our revocation scheme, which we will use to realize "negated" attributes in our ABE scheme, is based on the q-MEDDH assumption. The technique we use to deal with ordinary, non-negated attributes, is the same as [18], which was based on the BDDH assumption. To adapt that part to the q-MEDDH assumption, we note that the BDDH assumption is embedded (in many different ways) in the q-MEDDH assumption that we use. In the BDDH assumption, we are given $A = g^{\tilde{a}}, B = g^{\tilde{b}}, g^s$ and must distinguish $e(g,g)^{\tilde{a}\tilde{b}s}$ from a random element. We will implicitly set $\tilde{a} = \alpha/a_1^2$, and $\tilde{b} = a_1^2$. Note that in the q-MEDDH assumption, we are given $A = g^{\tilde{a}}$ and $B = g^{\tilde{b}}$ for these settings of $\tilde{a}$ and $\tilde{b}$. Below we will use A and B to mean these values.

Suppose there exists a polynomial-time adversary $\mathcal{A}$ that can attack our scheme in the selective-set model with advantage ϵ. We build a simulator $\mathcal{B}$ that can play the q-MEDDH game with advantage $\epsilon/2$. The simulation proceeds as follows:

The simulator begins by receiving a q-MEDDH challenge $\vec{X}, Z$. Note that with probability 1/2, $Z = e(g,g)^{\alpha s}$. We will denote this event as $\Xi = 0$. With probability 1/2, however, $Z = e(g,g)^z$ where z is a random element of $\mathbb{Z}_p$. We will denote this event as $\Xi = 1$.

Init The simulator $\mathcal{B}$ runs $\mathcal{A}$. $\mathcal{A}$ chooses the challenge set, γ, a set of d members of $\mathbb{Z}_p^*$.

Setup The simulator assigns the public parameters $g_1 = A$ and $g_2 = B$, thereby implicitly setting $\alpha' = \alpha/a_1^2$ and $\alpha'' = a_1^2$.

The simulator will also program the random oracle $H(x)$ as follows. Suppose the adversary queries the oracle on x. If the simulator already answered such a query, it simply returns the same answer. Otherwise, it picks a random $f_x \in \mathbb{Z}_p$ and responds as follows:

$$H(x) = \begin{cases} g^{f_x} & \text{if } x \in \gamma \\ g_2 g^{f_x} & \text{if } x \notin \gamma \end{cases}$$

The simulator sets up the remainder of the public key exactly as in the proof of the revocation scheme, where the revocation set $S^* = \gamma$.

Phase 1 $\mathcal{A}$ adaptively makes requests for several access structures such that γ passes through none of them. Suppose $\mathcal{A}$ makes a request for the secret key for an access structure $\tilde{\mathbb{A}}$ where $\tilde{\mathbb{A}}(\gamma) = 0$. Note that by assumption, $\tilde{\mathbb{A}}$ is given as $NM(\mathbb{A})$ for some monotonic access structure $\mathbb{A}$, over a set $\mathcal{P}$ of parties (whose names will be attributes), associated with a linear secret-sharing scheme Π.

Let M be the share-generating matrix for Π: Recall, M is a matrix over $\mathbb{Z}_p$ with ℓ rows and $n+1$ columns. For all $i = 1, \ldots, \ell$, the i'th row of M is labeled with a party named $\breve{x}_i \in \mathcal{P}$, where x_i is the attribute underlying $\breve{x}_i$. Note that $\breve{x}_i$ can be primed (negated) or unprimed (non-negated). When we consider the column vector $v = (s, r_1, r_2, \ldots, r_n)$, where s is the secret to be shared, and $r_1, \ldots, r_n \in \mathbb{Z}_p$ are randomly chosen, then Mv is the vector of ℓ shares of the secret s according to Π.

We make use of the following well-known observation about linear secret-sharing schemes (see, e.g. [2][4]): If $S \subset \mathcal{P}$ is a set of parties, then these parties can reconstruct the secret iff the column

[4] Here, we are essentially exploiting the equivalence between linear secret-sharing schemes and monotone span programs, as proven in [2]. The proof in [2] is for a slightly different formulation, but applies here as well.

vector $(1, 0, 0, \ldots, 0)$ is in the span of the rows of M_S, where M_S is the submatrix of M containing only those rows that are labeled by a party in S. Note that since $\tilde{\mathbb{A}}(\gamma) = 0$, we know that $\mathbb{A}(\gamma') = 0$, where $\gamma' = N(\gamma)$. Thus, we know that $(1, 0, \ldots, 0)$ is linearly independent of the rows of $M_{\gamma'}$.

During key generation, a secret sharing of the secret $\alpha' = \tilde{a}$ is supposed to be selected. In this simulation, however, we will choose this sharing (implicitly) in a slightly different manner, as we describe now: First, we pick a uniformly random vector $v = (v_1, \ldots, v_{n+1}) \in \mathbb{Z}_p^{n+1}$. Now, we make use of the following simple proposition [1, 27] from linear algebra:

Proposition B.2. A vector π is linearly independent of a set of vectors represented by a matrix N if and only if there exists a vector w such that $Nw = \vec{0}$ while $\pi \cdot w = 1$.

Since $(1, 0, \ldots, 0)$ is independent of $M_{\gamma'}$, there exists a vector $w = (w_1, \ldots, w_{n+1})$ such that $M_{\gamma'} w = \vec{0}$ and $(1, 0, \ldots, 0) \cdot w = w_1 = 1$. Such a vector can be efficiently computed [1, 27]. Now we define the vector $u = v + (\tilde{a} - v_1)w$. (Note that u is distributed uniformly subject to the constraint that $u_1 = \tilde{a}$.) We will implicitly use the shares $\vec{\lambda} = Mu$. This has the property that for any λ_i such that $\breve{x}_i \in \gamma'$, we have that $\lambda_i = M_i u = M_i v$ has no dependence on $\tilde{a}$.

Now that we have established how to distribute shares to "parties", which map to negated or non negated attributes, we need to show how to generate the key material.

We first describe how to generate decryption key material corresponding to negated parties $\breve{x}_i = x'_i$. Note that by definition, $\breve{x}_i \in \gamma'$ if and only if $x_i \notin \gamma$.

- If $x_i \in \gamma$, then since $\breve{x}_i \notin \gamma'$, we have that λ_i may depend linearly on $\tilde{a}$, and in general $\lambda_i = \mu\tilde{a} + \theta$, for some known constants μ and θ. However, by the simulator's choices at setup, we can invoke the proof of the revocation scheme to generate the appropriate key material. Note that in our setting, the randomness r_i is the name of the randomness t_i from the revocation scheme, and x_i is the name of the identity ID_i. Furthermore, note that with our parameters, we have that $D_i^{(3)} = g_2^{\lambda_i} g^{b^2 r_i} = g^{\mu\alpha} g^{b^2 r_i} \cdot g^{\theta\alpha''}$. Note that $g^{\theta\alpha''}$ can be generated immediately from $g^{\alpha''} = g^{a_1^2}$ which is given as part of the q-MEDDH assumption. The remainder of the key material is generated exactly as specified in the proof of the revocation scheme (see also the remark following the key generation part of the proof).
- If $x_i \notin \gamma$, then since $\breve{x}_i \in \gamma'$, we have that λ_i is independent of any secrets and is completely known to the simulator. In this case, the simulator chooses $r_i \in \mathbb{Z}_p$ at random, and outputs the following:
$$D_i = (D_i^{(3)} = g_2^{\lambda_i + b^2 r_i}, D_i^{(4)} = g^{r_i b x_i} h^{r_i}, D_i^{(5)} = g^{-r_i})$$
Note that the simulator can compute all these elements using elements already computed as part of the computation of the public key (g^{b^2}, g^b, h).

We now describe how to give key material corresponding to non negated parties $\breve{x}_i = x_i$. The simulated key construction techniques for non negated parties is similar to previous work [18, 28].

- If $x_i \in \gamma$, then since λ_i has no dependence on any unknown secrets, we simply choose $r_i \in \mathbb{Z}_p$, and output $D_i = (D_i^{(1)} = g_2^{\lambda_i} \cdot H(x_i)^{r_i}, D_i^{(2)} = g^{r_i})$.
- If $x_i \notin \gamma$, then we work as follows: Let $g_3 = g^{\lambda_i}$. Note that the simulator can compute g_3

using A and g. Choose $r'_i \in \mathbb{Z}_p$ at random, and output the components of D_i as follows:

$$\begin{aligned} D_i^{(1)} &= g_3^{-f_{x_i}} (g_2 g^{f_{x_i}})^{r'_i} \\ D_i^{(2)} &= g_3^{-1} g^{r'_i} \end{aligned}$$

Claim B.3. The simulation above produces valid decryption keys, that are furthermore distributed identically to the decryption keys that would have been produced by the ABE scheme for the same public parameters.

PROOF:

We will establish this claim by a case analysis. For key material corresponding to negated parties $\breve{x}_i$, this has already been verified in the proof of the revocation scheme.

For key material corresponding to non negated parties $\breve{x}_i$:

- If $x_i \in \gamma$, then the simulation produces key material using the same procedure as the ABE scheme.
- If $x_i \notin \gamma$, then to see why the simulated key material is good, note that by our programming of the hash function $H(x)$ has a g_2 component for all $x_i \notin \gamma$. Now let $r_i = r'_i - \lambda_i$. Note that r_i is distributed uniformly over $\mathbb{Z}_p$ and is independent of all other variables except r'_i. Then,

$$\begin{aligned} D_i^{(1)} &= g_3^{-f_{x_i}} (g_2 g^{f_{x_i}})^{r'_i} \\ &= g^{-\lambda_i f_{x_i}} (g_2 g^{f_{x_i}})^{r'_i} \\ &= g_2^{\lambda_i} (g_2 g^{f_{x_i}})^{-\lambda_i} (g_2 g^{f_{x_i}})^{r'_i} \\ &= g_2^{\lambda_i} (g_2 g^{f_{x_i}})^{r'_i - \lambda_i} \\ &= g_2^{\lambda_i} H(x_i)^{r_i} \end{aligned}$$

  and

$$D_i^{(2)} = g_3^{-1} g^{r'_i} = g^{r'_i - \lambda_i} = g^{r_i}$$

□

Challenge The adversary $\mathcal{A}$, will submit two challenge messages M_0 and M_1 to the simulator. Let C denote $g^s g^{s'}$, where s' is chosen at random, and g^s is as provided by the q-MEDDH assumption. The simulator flips a fair binary coin ν, and returns an encryption of M_ν. The ciphertext is output as

$$E = \Big(\gamma, E^{(1)} = M_\nu Z, E^{(2)} = C, \{E_x^{(3)} = C^{f(x)}\}_{x \in \gamma}, \{E_x^{(4)}\}, \{E_x^{(5)}\}\Big)$$

where $\{E_x^{(4)}\}, \{E_x^{(5)}\}$ are constructed exactly as $C_{i,1}$ and $C_{i,2}$, respectively, in the proof of the revocation scheme.

If $\Xi = 0$ then $Z = e(g,g)^{\alpha s}$. Then by inspection, the ciphertext is a valid ciphertext for the message M_ν under the set γ.

Otherwise, if $\Xi = 1$, then $Z = e(g,g)^z$. We then have $E^{(1)} = M_\nu e(g,g)^z$. Since z is random, $E^{(1)}$ will be a random element of $\mathbb{G}_T$ from the adversary's viewpoint and the message contains no information about M_ν.

Phase 2 The simulator acts exactly as it did in Phase 1.

Guess $\mathcal{A}$ will submit a guess ν' of ν. If $\nu' = \nu$ the simulator will output $\Xi' = 0$ to indicate that it was given a valid q-MEDDH tuple; otherwise, it will output $\Xi' = 1$ to indicate it was given a random target element Z.

As shown above, the simulator's generation of public parameters and private keys is identical to that of the actual scheme.

In the case where $\Xi = 1$ the adversary gains no information about ν. Therefore, we have $\Pr[\nu \neq \nu' | \Xi = 1] = \frac{1}{2}$. Since the simulator guesses $\Xi' = 1$ when $\nu \neq \nu'$, we have $\Pr[\Xi' = \Xi | \Xi = 1] = \frac{1}{2}$.

If $\Xi = 0$ then the adversary sees an encryption of M_ν. The adversary's advantage in this situation is ϵ by assumption. Therefore, we have $\Pr[\nu = \nu' | \Xi = 0] = \frac{1}{2} + \epsilon$. Since the simulator guesses $\Xi' = 0$ when $\nu = \nu'$, we have $\Pr[\Xi' = \Xi | \Xi = 0] = \frac{1}{2} + \epsilon$.

The overall advantage of the simulator in the q-MEDDH game is $\frac{1}{2} \Pr[\Xi' = \Xi | \Xi = 0] + \frac{1}{2} \Pr[\Xi' = \Xi | \Xi = 1] - \frac{1}{2} = \frac{1}{2}(\frac{1}{2} + \epsilon) + \frac{1}{2}\frac{1}{2} - \frac{1}{2} = \frac{1}{2}\epsilon$. □