Broadcast Encryption with Both Temporary and Permanent Revocation

Dan Brownstein[1], Niv Gilboa[2], and Shlomi Dolev[1]

[1] Dept. of Computer Science, Ben-Gurion University of the Negev, Israel, `dolev,danbr@cs.bgu.ac.il`,

[2] Dept. of Communication Systems Engineering, Ben-Gurion University of the Negev, Israel `gilboan@bgu.ac.il`

Abstract. Broadcast encryption enables a sender to broadcast data that only an authorized set of users can decrypt and is therefore an essential component of secure content distribution. Public key broadcast encryption separates the roles of a key manager who provides keys to users and content providers who distribute content to users. This separation is useful for flexible content distribution and for simplifying the process of additional content providers joining the network. A content provider or key manager can control the authorized set of users by user revocation which has two types, temporary revocation and permanent revocation. A content provider sending a message can determine the set of users authorized for the message by using temporary revocation. A key manager can use permanent revocation to remove a user from the set of authorized users as a better alternative to temporarily revoking the user in all subsequent messages. In this paper we present the first public-key, broadcast encryption scheme that achieves both temporary and permanent revocation and has essentially optimal performance. The scheme combines and optimizes the broadcast encryption systems of Delerablée et al. (Pairing 2007) and Lewko et al. (Security and Privacy 2010) and is generically secure over groups that support bilinear maps.

Regular paper submission.
Eligible for best student paper award.

1 Introduction

In broadcast encryption a single broadcaster can send encrypted messages to a group of users so that only authorized users can decrypt the messages. Since the introduction of broadcast encryption by Fiat and Naor in [FN93] there has been a great deal of work, e.g. [CGI$^+$99,CMN99,GSW00,NNL01,DF02,GST04], and [BGW05,DPP07,GW09,NP10,LSW10] on extending the framework of broadcast encryption, improving its security and optimizing its performance.

One of the factors driving interest in broadcast encryption is its commercial importance in content distribution, e.g. television networks. Historically, such networks were developed and administered by a single broadcaster who distributed both content and keys to registered users. In this setting it is perfectly reasonable to use symmetric-key encryption in which the broadcaster holds all the keys of the receivers.

A more flexible system enables separation of the key distribution and content distribution functions. In this setting a single key manager generates and distributes keys, but multiple content providers can directly send encrypted content to users. The benefits of such an approach are lower barriers of entry for both key providers and content providers and potentially greater choice and lower cost for users. However, the separation of functions typically rules out symmetric-key encryption since the key manager would not want to share all the system's keys with a content provider. Public-key broadcast encryption [DF02,BGW05,DPP07,GW09,LSW10] solves this problem by separating the keys into a public key allowing a content provider to encrypt content and secret keys allowing each authorized user to decrypt content.

Broadcast encryption schemes differ in the way they determine authorized users. Upon joining the system a user is authorized to receive a subset of the distributed content. This authorization is enforced by the keys that the key manager provides to the user. The key manager can decide to expand the subset of the content for which the user is authorized by providing additional keys. However, reducing the user's authorization or completely revoking that authorization requires a revocation procedure that invalidates the user's decryption keys.

Revocation in broadcast encryption schemes can be divided into two types, temporary and permanent. In temporary revocation [NNL01,BGW05,GW09] and [LSW10] authorization is attached to a specific encrypted message and therefore revoking a user does not extend to subsequent messages. In permanent revocation [CGI$^+$99,CMN99,GSW00] and the third construction of [DPP07] the key manager revokes the authorization of a user preventing it from decrypting future messages. Permanent revocation can be simulated by temporary revocation in which the revoked user is temporarily revoked in each message. However, that approach suffers from two drawbacks. The first is an obvious performance penalty since the complexity of sending a message keeps growing as a function of historical revocations. The other is that when the roles of key management and content distribution are separate it may not be possible for a broadcaster to keep track of all the revoked users.

Most works on revocation for broadcast encryption limit their goals either to temporary revocation only or to permanent revocation only, often without explicitly stating the difference[3]. However, in practice both types of revocation are important. Permanent revocation is the consequence of a user canceling his subscription and is therefore a common feature of real-world broadcast encryption systems. A motivating example for temporary revocation is when a content provider distributes an encryption key for some premium content, e.g. a televised pay-per-view event, only to users who paid for the content. Subsequently the content is broadcast to all users in the system, but only the users who received the content key can decrypt it.

The security of broadcast encryption can be loosely defined as the property of non-authorized users being unable to decrypt ciphertexts and can be typically reduced to the security of a cryptographic primitive. Such primitives include any symmetric key encryption [CGI+99,CMN99,GSW00,GST04], Hierarchical Identity Based Encryption [DF02], several q-type assumptions[4] on bilinear maps [BGW05,DPP07,GW09] and a combination of the Bilinear Decisional Diffie-Hellman assumption and the Decisional Linear assumption [LSW10].

Security definitions for broadcast security differ in modeling the adversary. One feature of the adversary model is the number of users that the adversary may corrupt. Most broadcast encryption schemes assume that the adversary can control multiple users, possibly an unbounded number of them, and therefore require *collusion resistance*, i.e. that even a coalition of unauthorized users working together cannot decrypt ciphertexts. A second feature determines whether the adversary (and the associated security proof) is *adaptive* or is only *selective*. An adaptive adversary decides dynamically which users to corrupt while in the selective setting the adversary selects the set of corrupted users before the key manager sets system parameters.

The performance of broadcast encryption is measured by the size of the objects in the system and the time required to perform the algorithms in the scheme as a function of the n users in the system and the number of revoked users. The measured objects include encryption and decryption keys, ciphertext length and messages for user revocation, which are part of the ciphertext in the case of temporary revocation and are separate for permanent revocation.

The performance of different broadcast encryption schemes is sometimes difficult to compare because each optimizes different measures. For example, the simplest broadcast encryption scheme involves encrypting a plaintext message separately with each authorized user's symmetric/public key. In this scheme the encryption key, ciphertext length and time to perform encryption are $O(n-r)$ for n users in the system and r revoked users. However, all other measures are $O(1)$ and revocation is especially trivial for all users actually requiring *less*

[3] The work of Delerablée et al. [DPP07] is an exception, considering both types of revocation. However, it focuses almost exclusively on temporary revocation, stating and analyzing the permanent revocation scheme very briefly.

[4] A q-type assumption is a family hardness assumptions indexed by an integer q, which corresponds to the number of queries the adversary makes in the security proof.

work for the key manager and broadcaster. In contrast, two efficient schemes are the public-key, temporary revocation scheme of Lewko et al. [LSW10] and the symmetric-key, permanent revocation scheme, which is the third scheme, of [DPP07]. In both schemes the size of all keys is $O(1)$, while in [LSW10] the ciphertext size and encryption and decryption time are $O(r)$ for r temporarily revoked users and in [DPP07] the length of a permanent revocation message, the time to construct the permanent revocation message and the time to update each secret user key are all $O(r')$ for r' permanently revoked users. An immediate implication is that if it is critical to minimize the running time of user devices then the simple broadcast encryption scheme is sufficient while if communication complexity and the key manager's workload are more important then other schemes such as [DPP07,LSW10] are preferable.

1.1 Contribution

The main contribution of this work is a public-key, broadcast encryption scheme that enables both temporary and permanent revocation with performance that in every measure is as good as the best broadcast encryption systems that achieve either temporary revocation or permanent revocation separately. At a high level we define a broadcast encryption scheme with temporary and permanent revocation as a protocol between a *key manager*, *n receivers* (or *users*) and an unbounded number of *broadcasters*. The protocol includes six algorithms: setup, key generation, encryption, decryption, (permanent) revocation and key update.

The key manager runs setup to generate system parameters including a master key, which it retains, and a public key which is published. The key manager also performs key generation to create a secret key for each user in the system. It is assumed that a user receives the secret key in a secure, out-of-band method, e.g. by VPN between the key manager and the user. A broadcaster executes the encryption algorithm which takes a set of temporarily revoked users as one of its parameters and outputs a ciphertext. A user can decrypt this ciphertext if and only if it is not one of the temporarily revoked users. The key manager performs the revocation algorithm which enables each of the non-revoked users to run key update and derive new secret keys. The revoked users will not be able to update their keys and will be unable to decrypt any ciphertexts in the future. However, it is always possible for a user to go through the key generation process again, receiving fresh keys.

The scheme combines ideas from the public-key, temporary revocation system of [LSW10] and the symmetric-key, permanent revocation suggested in [DPP07]. A seemingly attractive approach is to paste the two systems together in the sense of having each user hold independent keys for each system. A broadcaster secret shares each message and encrypts one share with the temporary revocation system and the other share with the permanent revocation system. Then a legitimate user can decrypt both shares and a revoked user will be unable to decrypt. However, this approach is insecure when considering collusion between users who are only temporarily revoked and users who are only permanently revoked.

As an alternative to pasting, our construction merges the keys of the two schemes and modifies the six algorithms appropriately to ensure correctness. We prove that the construction is secure based on a two-step hybrid argument. The first step relies on the hardness of a novel computational problem which is formalized as the Mixed Exponent Assumption (MEA) and the second step is secure in the generic bilinear group model [Sho97,BBG05].

MEA is similar to discrete-log based hardness assumptions in that its input includes powers of a generator g of a group $\mathbb{G}$ with prime order p. The group elements in MEA (as in other hardness assumptions) are not independently chosen. There are correlations between the exponents of different group elements. Unlike other assumptions, the input of MEA also includes functions of these exponents directly in $\mathbb{Z}_p$. Intuitively, any combination of the elements in $\mathbb{G}$and the elements in $\mathbb{Z}_p$ is indistinguishable from a random element. In slightly more detail, MEA states that the following problem is computationally hard. Let $Y \in \mathbb{Z}_p^m$ be a vector of secrets, let $V, U \in \mathbb{Z}_p^n$, let A be a $n \times m$ matrix, let F be a sequence of k multivariate polynomials over Y and let $H \in \mathbb{G}^k$. Then, it is hard to distinguish between the pairs (V, H) and (U, H) if:

- $V = AY$, U is randomly sampled from $\mathbb{Z}_p^n$, $H = g^{F(Y)}$, i.e. if $H = (h_1,\ldots,h_k)$, $F = (f_1, \ldots, f_k)$ and $Y = (y_1, \ldots, y_m)$ then $h_i = g^{f_i(y_1,\ldots,y_m)}$.
- There does not exist a low-degree, multivariate polynomial p over $\mathbb{Z}_p$ such that $p(a_1 f_1(x_1, \ldots, x_m), \ldots, a_k f_k(x_1, \ldots, x_m))$ is the zero polynomial, where $a_1, \ldots, a_k$ are arbitrary functions of $V = (v_1, \ldots, v_n)$. If $\mathbb{G}$ comes equipped with a bilinear mapping then p must be of degree two, while otherwise it must be linear.

The MEA assumption is in fact a family of assumptions indexed by a matrix and a sequence of functions. We refer to a specific member of this family with matrix A and sequence of functions F as (A, F)-MEA.

Our construction has similar performance to a combination of the performance of [DPP07] and [LSW10]. The public key and each secret key are of size $O(1)$ group elements. A ciphertext which determines the temporary revocation of r users is of length $O(r)$ group elements and the time complexity of both encryption and decryption is $O(r)$. Similarly, the output of the revocation algorithm, which is used for permanent revocation of r' users is of length $O(r')$ and the time complexity of both the revocation and key update algorithms are $O(r')$.

2 Preliminaries

2.1 Revocation Systems

A revocation scheme that supports both temporary and permanent revocations consists of six algorithms: Setup, KeyGen, Revoke, UpdateKey, Encrypt and Decrypt.

$\mathsf{Setup}(\lambda)$. The setup algorithm takes as input the security parameter λ and outputs public parameters PP and a master secret key MSK.

$\mathsf{KeyGen}(MSK, ID)$. The key generation algorithm takes as input the master secret key MSK and an identity ID and outputs a secret key SK_{ID}. Each key has a boolean property $SK_{ID}.\mathsf{revoked}$ which is set by default to false.

$\mathsf{Revoke}(S, PP, MSK)$. The revocation algorithm takes as input the master secret key MSK, the public parameters PP and a set S of identities to revoke. The algorithm outputs a new master secret MSK', new public parameters PP' and a key update message SUM. PP' and SUM are broadcast to all users.

$\mathsf{UpdateKey}(SK_{ID}, SUM, ID)$. The key update algorithm takes as input the user's secret key SK_{ID}, the key update message SUM and the user's identity ID. The algorithm outputs a new secret key SK'_{ID}. If ID is in the set of revoked users that corresponds to SUM, the algorithm sets $SK'_{ID}.\mathsf{revoked} = \mathsf{true}$.

$\mathsf{Encrypt}(S, PP, M)$. The encryption algorithm takes as input a set S of identities to revoke, the public parameters PP and a message M. The algorithm outputs a ciphertext CT.

$\mathsf{Decrypt}(SK_{ID}, CT, PP)$. The decryption algorithm takes as input a secret key, SK_{ID}, a ciphertext CT and the public parameters PP. If $SK_{ID}.\mathsf{revoked} = \mathsf{true}$ or ID is in the set of revoked users that corresponds to CT, the algorithm outputs $\perp$. Otherwise it outputs the message M associated with CT.

The system must satisfy the following correctness and security properties.

Correctness. For all messages M, sets of identities $S, S_1 \ldots, S_n$ and all $ID \notin \bigcup_{i=1}^{n} S_i \cup S$, if $(PP_0, MSK_0) \leftarrow \mathsf{Setup}(\lambda)$, $SK_{ID,0} \leftarrow \mathsf{KeyGen}(MSK, ID)$ and for $i = 1, \ldots, n$:

$$\begin{array}{cl} (MSK_i, PP_i, SUM_i) & \leftarrow \mathsf{Revoke}(S_i, PP_{i-1}, MSK_{i-1}), \\ SK_{ID,i} & \leftarrow \mathsf{UpdateKey}(SK_{ID,i-1}, SUM_i, ID) \end{array}$$

then if $CT \leftarrow \mathsf{Encrypt}(S, PP_n, M)$ then $\mathsf{Decrypt}(SK_{IDn}, CT, PP_n) = M$.

Security. The security of a permanent revocation scheme is defined as a game between a challenger and an attack algorithm $\mathcal{A}$ with the following phases:

Setup. The challenger runs the *Setup* algorithm with security parameter λ to obtain the public parameters PP and the master secret key MSK. It maintains a set of identities Q initialized to the empty set and then sends PP to $\mathcal{A}$.

Key Query and Revocation. In this phase $\mathcal{A}$ adaptively issues secret key and revocation queries. For every private key query for identity ID, the challenger adds ID to Q, runs $\mathsf{KeyGen}(MSK, ID) \rightarrow SK_{ID}$ and sends $\mathcal{A}$ the corresponding secret key SK_{ID}. For every revocation query for a set S of Identities, the challenger updates $Q \leftarrow Q \setminus S$, runs $\mathsf{Revoke}(S, PP, MSK) \rightarrow (MSK', PP', SUM)$, replaces (MSK, PP) with (MSK', PP') and sends $\mathcal{A}$ the new PP and the corresponding key update messages SUM.

Challenge. $\mathcal{A}$ sends the challenger a set S of identities and two messages M_1, M_2. In case $Q \nsubseteq S$ the challenger sends $\perp$ to $\mathcal{A}$ and aborts. Otherwise, the challenger flips a random coin $b \in \{0, 1\}$, runs the $\mathsf{Encrypt}(S, PP, M_b)$ algorithm to obtain an encryption of M_b and sends it to $\mathcal{A}$.

Guess. $\mathcal{A}$ outputs a guess $b' \in \{0, 1\}$ and wins if $b = b'$.

The advantage $\mathcal{A}$ has in the security game for a permanent revocation scheme with security parameter λ is defined as

$$Adv_{\mathcal{A},\lambda} = \left| Pr[\mathcal{A} \text{ wins}] - \frac{1}{2} \right|$$

A permanent revocation scheme is adaptively secure if for all poly-time algorithms $\mathcal{A}$ we have that $Adv_{\mathcal{A},\lambda} = negl(\lambda)$.

We note that selective security is defined similarly, except that the revoked sets of identities are declared by the adversary before it sees the public parameters in an Init phase.

2.2 Bilinear maps

For groups $\mathbb{G}, \mathbb{G}_T$ of the same prime order p, a bilinear map $e : \mathbb{G}^2 \rightarrow \mathbb{G}_T$ satisfies:

1. Bilinearity. For every $g_1, g_2 \in \mathbb{G}$ and $\alpha \in \mathbb{Z}_p$ it holds that
$$e(g_1^\alpha, g_2) = e(g_1, g_2^\alpha) = e(g_1, g_2)^\alpha.$$
2. Non-degeneracy. If $g_1, g_2 \in \mathbb{G}$ are generators of $\mathbb{G}$ then $e(g_1, g_2)$ is a generator of $\mathbb{G}_T$.

We call $\mathbb{G}$ a (symmetric) bilinear group and $\mathbb{G}_T$ the target group.

2.3 Pseudo-Random Functions

Intuitively, a family of Pseudo-Random Functions (PRF) [GGM84,NR04] is a family of functions such that a randomly chosen member of the family cannot be efficiently distinguished from a truly random function by an algorithm that observes the function's output. The following definition states the requirements of a PRF more precisely for the systems we construct.

Definition 1. *Let $(A_\lambda, B_\lambda)_{\lambda \in \mathbb{N}}$ be a sequence of pairs of domains and let $F = \{F_\lambda\}_{\lambda \in \mathbb{N}}$ be a function ensemble such that the random variable F_λ assumes values in the set of $A_\lambda \rightarrow B_\lambda$ functions. Then F is a PRF ensemble if:*

- *Pseudo-randomness. For every PPT oracle machine M*

$$|Pr(M^{F_\lambda}(1^\lambda) = 1 - Pr(M^{U_\lambda}(1^\lambda) = 1| < negl(\lambda),$$

for a negligible function $negl(\cdot)$ where U_λ is distributed uniformly over $A_\lambda \to B_\lambda$ functions.
- *Efficient computation There are efficient PPT algorithms to sample F_λ and compute the sampled function on any input.*

The constructions of [GGM84,NR04] show that PRF function families can be efficiently constructed for many useful domains A_λ, B_λ including the groups $\mathbb{Z}_n = \mathbb{Z}/n\mathbb{Z}$ for any integer n. Note that in practice a PRF family can be instantiated by a block cipher such as AES where each function in the family corresponds to a cipher key.

2.4 The Decisional MEA-assumption

Let $\mathbb{G}$ be a group of prime order p with generator g, let $A \in \mathbb{Z}_p^{n\times m}$ be matrix and for a vector of formal variables $X = (X_1, \ldots, X_m)$ define a vector $Y = (Y_1, \ldots, Y_n)$ by $AX = Y$. Let $F = (f_1, \ldots, f_k) \in \mathbb{Z}_p[X_1, \ldots, X_m]^k$ be a k-tuple of m-variate polynomials such that for any degree two, k-variate polynomial p and for any k polynomials $\phi_1, \ldots, \phi_k \in \mathbb{Z}_p[Y_1, \ldots, Y_n]^k$ the polynomial $p(\phi_1(Y)f_1(X), \ldots, \phi_k(Y)f_k(X))$ is not identically zero.

The Decisional (A, F)-Mixed Exponent problem in $\mathbb{G}$ is to distinguish between the distributions on the tuples $(g, \mathbf{y}, \mathbf{h})$ and $(g, \mathbf{u}, \mathbf{h})$ which are defined as follows. Choose a random $\mathbf{x} = (x_1, \ldots, x_m) \in \mathbb{Z}_p^m$, compute $\mathbf{y} = A \cdot \mathbf{x}$ and $\mathbf{h} = (h_1, \ldots, h_k)$ such that $h_i = g^{f_i(x_1,\ldots,x_m)}$ for all i and sample $\mathbf{u} \in \mathbb{Z}_p^n$ randomly.

An algorithm $\mathcal{D}$ that outputs $z \in \{0, 1\}$ has advantage ϵ in solving the Decisional (A, F)-MEA-problem in $\mathbb{G}$ if

$$\left| Pr[\mathcal{D}(g, A, F, \mathbf{y}, \mathbf{h}) = 1] - Pr[\mathcal{D}(g, A, F, \mathbf{u}, \mathbf{h}) = 1] \right| \geq \epsilon$$

We say that the Decisional (A, F)-Mixed Exponent Assumption ((A, F)-MEA) holds if any poly-time algorithm has a negligible advantage in solving the Decisional (A, F)-MEA-problem. The MEA family of assumptions is the set of all (A, F)-MEA-assumptions.

3 Public Key Revocation Scheme

$\mathsf{Setup}(\lambda)$. The setup algorithm, given a security parameter λ, chooses a bilinear group $\mathbb{G}$ of prime order p such that $|p| \geq \lambda$. It then chooses random generators $g, w \in \mathbb{G}$, random exponents $\alpha, \gamma, b \in \mathbb{Z}_p$ and sets $ST = 1$. Finally, the setup algorithm randomly chooses a function ϕ[5] from F_λ, a pseudo-random family of permutations over $\mathbb{Z}_p$.

The master secret key is

$$MSK = (\alpha, b, \gamma, w, ST, \phi)$$

And the public parameters are

$$PP = (g, g^{bST}, g^{b^2 ST}, w^b, e(g,g)^{\alpha ST})$$

$\mathsf{KeyGen}(MSK, ID)$. Given a user identity $ID \in \mathbb{Z}_p$ and the master secret key MSK, the algorithm computes $t = \phi(ID) \in \mathbb{Z}_p$ and sets:

$$D_1 = g^{-t}, D_2 = (g^{bID}w)^t,$$
$$D_3 = \frac{1}{\alpha + b^2 t} - \gamma, D_4 = g^{(\alpha+b^2 t)\cdot ST}$$
$$D_5 = \text{false}$$

The output of the algorithm is $SK_{ID} = \{D_1, \dots, D_5\}$.

Revoke(S, PP, MSK). The algorithm is given a set $S = \{ID_1, \dots, ID_r\}$ of identities to revoke, the public parameters and the master secret key. The algorithm sets $ST' = ST$ and for $i = 1$ to r it computes:

1. $ST' = ST' \cdot (\alpha + b^2 t_i)$
2. $S_{i,1} = \frac{1}{\alpha + b^2 t_i} - \gamma, S_{i,2} = g^{ST'}$

where $t_i = \phi(ID_i)$. The algorithm then:

1. Updates the master secret key by replacing ST with ST'.
2. Updates the public parameters by replacing $g^{bST}, g^{b^2 ST}$ and $e(g,g)^{\alpha ST}$ with $g^{bST'}, g^{b^2 ST'}$ and $e(g,g)^{\alpha ST'}$ respectively.
3. Broadcasts the key update message $SUM = \{S_{i,1}, S_{i,2}\}_{i=1}^{r}$.

UpdateKey(SK_{ID}, SUM, ID). Given a key update message SUM for r revoked identities, the algorithm updates the secret key SK_{ID}. It first checks if $D_3 \in \bigcup_{i=1}^{r} S_{i,1}$ and if so it sets $D_5 = \text{true}$. Otherwise, it sets $h_0 = D_4$. Then, for $i = 1$ to r it sets $h_i = \left(\frac{S_{i,2}}{h_{i-1}}\right)^{D_3 - S_{i,1}}$. Finally, the algorithm updates SK_{ID} by replacing D_4 with h_r.

We note that $h_r = g^{(\alpha+b^2 t)\cdot ST}$ where ST is the new state in the master secret key after the corresponding revocation. For example, if $t = \phi(ID)$ and

[5] We slightly abuse notation and use ϕ to denote both the function and a concrete description of this function.

$\hat{t} = \phi(\hat{ID})$, then the update process of SK_{ID} after the revocation of $\hat{ID}$ is

$$\begin{aligned} h_1 = \left(\frac{S_{1,2}}{h_0}\right)^{D_3 - S_{1,1}} &= \left(\frac{g^{\alpha+b^2\hat{t}}}{g^{\alpha+b^2 t}}\right)^{\frac{1}{\left(\frac{1}{\alpha+b^2 t}-\gamma\right)-\left(\frac{1}{\alpha+b^2\hat{t}}-\gamma\right)}} \\ &= \left(\frac{g^{\alpha+b^2\hat{t}}}{g^{\alpha+b^2 t}}\right)^{\frac{1}{\frac{(\alpha+b^2\hat{t})-(\alpha+b^2\hat{t})}{(\alpha+b^2 t)(\alpha+b^2\hat{t})}}} \\ &= g^{(\alpha+b^2 t)(\alpha+b^2\hat{t})} \end{aligned}$$

Encrypt(S, PP, M)*.* The encryption algorithm takes as input the public parameters PP, a message $M \in \mathbb{G}_T$ and a set S of revoked identities. The algorithm first lets $r = |S|$, chooses r random exponents $s_1, \ldots, s_r \in \mathbb{Z}_p$ and sets $s = \sum\limits_{i=1}^{r} s_i$. Next, the algorithm sets

$$C_0 = M \cdot e(g,g)^{\alpha s ST}, C_1 = g^s$$

and for $i = 1$ to r it sets

$$C_{i,1} = ID_i, C_{i,2} = (g^{bST})^{s_i}, C_{i,3} = (g^{b^2 STID_i} w^b)^{s_i}$$

The output of the algorithm is $CT = \{C_0, C_1, \{C_{i,1}, C_{i,2}, C_{i,3}\}_{i=1}^{r}\}$.

Decrypt(SK_{ID}, CT, PP)*.* The algorithm is given a secret key SK_{ID}, a ciphertext CT and the public parameters PP. First, if $D_5 =$ true or $ID \in \bigcup\limits_{i=1}^{r} C_{i,1}$ the algorithm outputs $\perp$. Otherwise the algorithm calculates:

$$\begin{aligned} A &= e(C_1, D_4) \\ &= e(g^s, g^{(\alpha+b^2 t)\cdot ST}) \\ &= e(g,g)^{\alpha s ST} \cdot e(g,g)^{b^2 stST} \\ B &= \prod_{i=1}^{r} \left(e(C_{i,2}, D_2) \cdot e(C_{i,3}, D_1)\right)^{\frac{1}{ID - C_{i,1}}} \\ &= \prod_{i=1}^{r} \left(e((g^{bST})^{s_i}, (g^{bID} w)^t) \cdot e((g^{b^2 STID_i} w^b)^{s_i}, g^{-t})\right)^{\frac{1}{ID-ID_i}} \\ &= e(g,g)^{\sum_{i=1}^{r} b^2 s_i tST} = e(g,g)^{b^2 stST} \end{aligned}$$

Finally the algorithm retrieves the message

$$M = C_0/(A/B)$$

4 Security Analysis

In this section we prove the following theorem:

Theorem 1. *Suppose the decisional MEA assumption holds. Then no poly-time adversary can break our revocation scheme in the generic group model.*

We prove that our revocation scheme is generically secure over bilinear groups by simulating a sequence of the security games against an adversary, each one indistinguishable from the next. The first game is the original security game. In the next game we replace the elements correspond to D_3 in the secret keys with random elements. Indistinguishably follows directly from the MEA assumption. In the next game we change the cipher text to a random element by using the framework of [BBG05].

Given an adversary $\mathcal{A}$ with non-negligible advantage $\epsilon = \mathrm{ADV}_{\mathcal{A}}$ in the security game against our construction we define a simulator $\mathcal{B}$ that plays the decisional MEA problem.

Let $A \in \mathbb{Z}_p^{n\times(2n+q+4)}$ be the following matrix

$$A^{n\times(2n+4+q)} = \left(\begin{array}{c|ccc|c} -1 & 0 & \cdots & 0 & \\ \cline{2-4} -1 & & & & \\ \vdots & & I^{n\times n} & & \mathbf{0} \\ -1 & & & & \end{array}\right)$$

Let $(p'_1, \ldots, p'_7) \in \mathbb{F}_p[Y_1, \ldots, Y_m]^7$ be the following polynomials

$$\begin{aligned}
&p'_1(Y_1) = Y_1^{-1}\\
&p'_2(y_1, Y_2, Y_3) = (Y_1^{-1} - Y_2^{-1}) * Y_3^2\\
&p'_3(Y_1) = Y_1^{-2}\\
&p'_4(Y_1, Y_2) = Y_1^{-1} * Y_2^{-1}\\
&p'_5(Y_1, Y_2, Y_3, Y_4, Y_5) = (Y_2^{-1} * Y_4 + Y_5^{-1})(Y_1^{-1} - Y_2^{-1}) * Y_3^2\\
&p'_6(Y_1, \ldots, Y_q) = \sum_{i=1}^{q} Y_i^{-1}\\
&p'_7(Y_1, Y_2, Y_3, Y_4) = (Y_1^{-2} * Y_2 + Y_3^{-1})(Y_4^{-1})
\end{aligned}$$

Let $F = (f_1, \ldots, f_k) \in \mathbb{F}_p[X_1, \ldots, X_m]^k$ be the following polynomials

$$\begin{array}{ll}
\forall_{1\le i\le n} & f_{y,i} = p'_1(X_{i+1})\\
\forall_{1\le i\le n} & f_{t1,i} = p'_2(X_{i+1}, X_{n+2}, X_{n+3})\\
 & fb = p'_1(X_{n+3})\\
 & fb2 = p'_3(X_{n+3})\\
 & fwb = p'_4(X_{n+3}, X_{n+4})\\
\forall_{1\le i\le n} & f_{t2,i} = p'_5(X_{i+1}, X_{n+2}, X_{n+3}, X_{n+4+i}, X_{n+4})\\
 & ps = f'_6(X_{2n+5}, \ldots, X_{2n+q+4})\\
\forall_{1\le i\le q} & f_{s1,i} = p'_4(X_{n+3}, X_{2n+4+i})\\
\forall_{1\le i\le q} & f_{s2,i} = p'_7(X_{n+3}, X_{n+4+i}, X_{n+4}, X_{2n+4+i})\\
 & f\alpha s = p'_6(X_{2n+5}, \ldots, X_{2n+q+4}) p'_1(X_{n+2})
\end{array}$$

$\mathcal{B}$ receives an (A,F)-MEA challenge $\mathbf{X} = (g, A, F, T, \mathbf{h})$. It then play the original security game against $\mathcal{A}$. Let τ be the number of permanent revocation requests that the adversary performs. Let ρ_i denote the number of revoked users in the i-th request. We denote their identities by ID_{i_j} where i is in $[1, \tau]$ and j is in $[1, \rho_i]$. Similarly, we use $ST_{i,j}$ to denote the state after the revocation of the j-th identity in the i-th group. Let ψ_i denote the number of secret key requests the adversary performs after the i-th permanent revocation request (ψ_0 is the number of secret key requests prior to the first revocation). We denote the identities for which the adversary requests keys by ID_{k_m} where k is in $[0, \tau]$ and m is in $[1, \psi_i]$ and $t_{k,m}$ to denote $f(ID_{k_m})$. Let q denote the number of users the adversary revoke during the temporary revocation. We denote their identities by ID_i where i in $[1, q]$.

We next write the elements that the adversary learns during the security game from which we state a computational assumption. From the public parameters and revocation requests, the adversary learns

$$\forall_{i\in[0,\tau],j\in[1,\rho_i]} \qquad \frac{1}{ST_{i,j}} - \gamma, g^{ST_{i,j}}, g^{b\cdot ST_{i,j}}, g^{b^2\cdot ST_{i,j}}, e(g,g)^{\alpha\cdot ST_{i,j}}$$

where

$$ST_{i,j} = \prod_{i'=1}^{i} \prod_{j'=1}^{j} (\alpha + b^2 t_{i'_{j'}})$$

From the secret key requests, the adversary learns

$$\forall_{k\in[0,\tau],m\in[1,\psi_k]} \quad ID_{k_m}, g^{-t_{k_m}}, (g^{bID_{k_m}} w)^{t_{k_m}}, \frac{1}{\alpha + b^2 t_{k_m}} - \gamma, g^{(\alpha+b^2 t_{k_m})ST_{k,m}}$$

where

$$ST_{k,m} = \prod_{k'=1}^{k} \prod_{m'=1}^{\rho'_k} (\alpha + b^2 t_{k'_{m'}})$$

Finally, from the challenge, the adversary learns

$$\begin{array}{ll} & g^s, M \cdot e(g,g)^{\alpha s ST_{final}} \\ \forall_{i\in[1,q]} & (g^{bST_{final}})^{s_i}, (g^{b^2 ST_{final} ID_i} w^b)^{s_i} \end{array}$$

where

$$ST_{final} = \prod_{i=1}^{\tau} \prod_{j=1}^{\rho_i} (\alpha + b^2 t_{i_j})$$

For each identity ID_{k_m} the adversary obtains a key we have that either ID_{k_m} is revoked in one of the $(\tau - k)$ permanent revocations following the creation of $SK_{ID_{k_m}}$, or that ID_{k_m} is revoked in the temporary revocation during the challenge phase. Thus, the next assumption captures the security of our scheme.

(n-q)-Decisional Multi-Exponent Assumption Let $\mathbb{G}$ be a bilinear group of prime order p. For any $(\tau, \rho_1, \ldots, \rho_\tau, \psi_0, \ldots, \psi_\tau)$ such that $\sum\limits_{k=0}^{\tau} \psi_k = n$ and $\sum\limits_{i=1}^{\tau} \rho_i = n - q$ the (n-q)-Decisional Multi-Exponent problem in $\mathbb{G}$ is as follows:

A challenger chooses generators $g, w \in \mathbb{G}$ and random exponents $\alpha, b, \gamma, \{t_{k_m}\}_{k\in[0,\tau],m\in[1,\psi_k]}$. Suppose an adversary is given $\mathbf{X} =$

$$\begin{array}{ll}
\forall_{i\in[0,\tau],j\in[1,\rho_i]} & \begin{cases} ID_{i_j}, \\ \frac{1}{\prod\limits_{i'=1}^{i}\prod\limits_{j'=1}^{j}(\alpha+b^2 t_{{i'}_{j'}})} - \gamma, g^{\prod\limits_{i'=1}^{i}\prod\limits_{j'=1}^{j}(\alpha+b^2 t_{{i'}_{j'}})}, \\ g^{b\cdot\prod\limits_{i'=1}^{i}\prod\limits_{j'=1}^{j}(\alpha+b^2 t_{{i'}_{j'}})}, g^{b^2\cdot\prod\limits_{i'=1}^{i}\prod\limits_{j'=1}^{j}(\alpha+b^2 t_{{i'}_{j'}})}, \\ e(g,g)^{\alpha\cdot\prod\limits_{i'=1}^{i}\prod\limits_{j'=1}^{j}(\alpha+b^2 t_{{i'}_{j'}})} \end{cases} \\
\forall_{k\in[0,\tau],m\in[1,\psi_k]} & \begin{cases} ID_{k_m}, g^{-t_{k_m}}, (g^{bID_{k_m}}w)^{t_{k_m}}, \frac{1}{\alpha+b^2 t_{k_m}} - \gamma, \\ g^{(\alpha+b^2 t_{k_m})\cdot\prod\limits_{k'=1}^{k}\prod\limits_{m'=1}^{\rho'_k}(\alpha+b^2 t_{{k'}_{m'}})} \end{cases} \\
 & g^s \\
\forall_{\ell\in[1,q]} & (g^{b\cdot\prod\limits_{i=1}^{\tau}\prod\limits_{j=1}^{\rho_i}(\alpha+b^2 t_{i_j})})^{s_\ell}, (g^{b^2\cdot\prod\limits_{i=1}^{\tau}\prod\limits_{j=1}^{\rho_i}(\alpha+b^2 t_{i_j})\cdot ID_\ell} w^b)^{s_\ell}
\end{array}$$

such that

$$\{ID_{k_m}\}_{k\in[0,\tau],m\in[1,\psi_k]} \setminus \left(\{ID_{i_j}\}_{i\in[0,\tau],j\in[1,\rho_i]} \cup \{ID_\ell\}_{\ell\in[1,q]}\right) = \emptyset$$

Then it must be hard to distinguish

$$T = e(g,g)^{\alpha s\cdot\prod\limits_{i=1}^{\tau}\prod\limits_{j=1}^{\rho_i}(\alpha+b^2 t_{i_j})}$$

from a random element $R \in \mathbb{G}_T$. An algorithm $\mathcal{A}$ that outputs $z \in \{0,1\}$ has advantage ϵ in solving the (n-q)-Decisional Multi-Exponent problem in $\mathbb{G}$ if

$$Adv^{\text{nqdme}}(n,q,\mathcal{A}) := \left| Pr[\mathcal{A}(\mathbf{X}, T = e(g,g)^{\alpha s\cdot\prod\limits_{i=1}^{\tau}\prod\limits_{j=1}^{\rho_i}(\alpha+b^2 t_{i_j})})] - Pr[\mathcal{A}(\mathbf{X}, T = R)] \right| \geq \epsilon$$

We say that the decision (n-q)-Decisional Multi-Exponent Assumption holds if no poly-time algorithm has a non-negligible advantage in solving the (n-q)-Decisional Multi-Exponent problem.

We next wish to use the proof template of Boneh et al. [BBG05] to show that the (n-q)-Decisional Multi-Exponent assumption is generically secure. Unfortunately, our assumption is outside their framework since it involves elements which are outside of the bilinear or the target groups. Int particular, the adversary is given elements of the form $\frac{1}{\prod_{i'=1}^{i}\prod_{j'=1}^{j}(\alpha+b^2 t_{i'_{j'}})} - \gamma \in \mathbb{Z}_p$.

In the second game, $\mathcal{B}$ uses the (A,F)-MEA challenge to simulate the security game. We denote by $\mathbf{x} = (\gamma, y_1, \ldots, y_n, \frac{1}{\alpha}, \frac{1}{b}, \frac{1}{w}, ID_1, \ldots, ID_n, \frac{1}{s_1}, \ldots, \frac{1}{s_q})$ the random exponents chosen by the (A,F)-MEA challenger. $\mathcal{B}$ can use $\mathbf{h}$ to simulate all the needed the groups elements since

$$\begin{aligned}
\forall_{1\le i\le n} \qquad & g^{p_{y,i}(\mathbf{x_{i+1}})} = g^{\alpha+b^2 t_i} \\
\forall_{1\le i\le n} \qquad & g^{p_{t1,i}(\mathbf{x_{i+1}},\mathbf{x_{n+2}},\mathbf{x_{n+3}})} = g^{-t_i} \\
& g^{pb(\mathbf{x_{n+3}})} = g^{b} \\
& g^{pb2(\mathbf{x_{n+3}})} = g^{b^2} \\
& g^{pwb(\mathbf{x_{n+3}},\mathbf{x_{n+4}})} = w^{b} \\
\forall_{1\le i\le n} \qquad & g^{p_{t2_i}(\mathbf{x_{i+1}},\mathbf{x_{n+2}},\mathbf{x_{n+3}},\mathbf{x_{n+4+i}},\mathbf{x_{n+4}})} = g^{(bID_i+b)t_i} \\
& g^{ps(\mathbf{x_{2n+5}},\ldots,\mathbf{x_{2n+q+4}})} = g^{s} \\
\forall_{1\le i\le q} \qquad & g^{p_{s1_i}(\mathbf{x_{n+3}},\mathbf{x_{2n+4+i}})} = g^{bs_i} \\
\forall_{1\le i\le q} \qquad & g^{p_{s2,i}(\mathbf{x_{n+3}},\mathbf{x_{n+4+i}},\mathbf{x_{n+4}},\mathbf{x_{2n+4+i}})} = g^{(b^2ID_j+b)s_i} \\
& e(g,g)^{p\alpha s(\mathbf{x_{2n+5}},\ldots,\mathbf{x_{2n+q+4}}\mathbf{x_{n+2}})}) = e(g,g)^{\alpha s}
\end{aligned}$$

$\mathcal{B}$ takes D_3 elements from T. If $T = A \times \mathbf{e}$ these elements have the same distribution as in the real security game since $T_i = \frac{1}{\alpha+b^2 t_i} - \gamma$. It follows that the simulation provided by $\mathcal{B}$ is perfect unless T is a random element in $\mathbb{Z}_p^n$. Assuming the MEA holds, $\mathcal{A}$ cannot distinguish between the games. We finish the proof in appendix A by showing that the altered assumption is generically secure using the [BBG05] framework.

References

[BBG05] Dan Boneh, Xavier Boyen, and Eu-Jin Goh. Hierarchical identity based encryption with constant size ciphertext. *IACR Cryptology ePrint Archive*, 2005:15, 2005.

[BGW05] Dan Boneh, Craig Gentry, and Brent Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In *CRYPTO*, volume 3621 of *Lecture Notes in Computer Science*, pages 258–275. Springer, 2005.

[CGI+99] Ran Canetti, Juan A. Garay, Gene Itkis, Daniele Micciancio, Moni Naor, and Benny Pinkas. Multicast security: A taxonomy and some efficient constructions. In *INFOCOM*, pages 708–716. IEEE, 1999.

[CMN99] Ran Canetti, Tal Malkin, and Kobbi Nissim. Efficient communication-storage tradeoffs for multicast encryption. In *EUROCRYPT*, volume 1592 of *Lecture Notes in Computer Science*, pages 459–474. Springer, 1999.

[DF02] Yevgeniy Dodis and Nelly Fazio. Public key broadcast encryption for stateless receivers. In *Digital Rights Management Workshop*, volume 2696 of *Lecture Notes in Computer Science*, pages 61–80. Springer, 2002.

[DPP07] Cécile Delerablée, Pascal Paillier, and David Pointcheval. Fully collusion secure dynamic broadcast encryption with constant-size ciphertexts or decryption keys. In *Pairing*, volume 4575 of *Lecture Notes in Computer Science*, pages 39–59. Springer, 2007.

[FN93] Amos Fiat and Moni Naor. Broadcast encryption. In *CRYPTO*, volume 773 of *Lecture Notes in Computer Science*, pages 480–491. Springer, 1993.

[GGM84] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct randolli functions. In *Foundations of Computer Science, 1984. 25th Annual Symposium on*, pages 464–479. IEEE, 1984.

[GST04] Michael T. Goodrich, Jonathan Z. Sun, and Roberto Tamassia. Efficient tree-based revocation in groups of low-state devices. In *CRYPTO*, volume 3152 of *Lecture Notes in Computer Science*, pages 511–527. Springer, 2004.

[GSW00] Juan A. Garay, Jessica Staddon, and Avishai Wool. Long-lived broadcast encryption. In *CRYPTO*, volume 1880 of *Lecture Notes in Computer Science*, pages 333–352. Springer, 2000.

[GW09] Craig Gentry and Brent Waters. Adaptive security in broadcast encryption systems (with short ciphertexts). In *EUROCRYPT*, volume 5479 of *Lecture Notes in Computer Science*, pages 171–188. Springer, 2009.

[LSW10] Allison B. Lewko, Amit Sahai, and Brent Waters. Revocation systems with very small private keys. In *IEEE Symposium on Security and Privacy*, pages 273–285. IEEE Computer Society, 2010.

[NNL01] Dalit Naor, Moni Naor, and Jeffery Lotspiech. Revocation and tracing schemes for stateless receivers. In *CRYPTO*, volume 2139 of *Lecture Notes in Computer Science*, pages 41–62. Springer, 2001.

[NP10] Moni Naor and Benny Pinkas. Efficient trace and revoke schemes. *Int. J. Inf. Sec.*, 9(6):411–424, 2010.

[NR04] Moni Naor and Omer Reingold. Number-theoretic constructions of efficient pseudo-random functions. *Journal of the ACM (JACM)*, 51(2):231–262, 2004.

[Sho97] Victor Shoup. Lower bounds for discrete logarithms and related problems. In *Eurocrypt*, volume 97, pages 256–266. Springer, 1997.

A Generic security of (n-q)-Decisional Exponent Assumption

Using the terminology of [BBG05] our assumption is the following (P, Q, f) Diffie-Hellman assumption where we denote by ν_w the discrete log of w in base g.

$$
\begin{aligned}
P = \{1, s\}& \\
\cup \{\forall_{i\in[0,\tau],j\in[1,\rho_i]} \quad & \prod_{i'=1}^{i}\prod_{j'=1}^{j}(\alpha + b^2 t_{i'_{j'}}), b\cdot\prod_{i'=1}^{i}\prod_{j'=1}^{j}(\alpha + b^2 t_{i'_{j'}}), b^2\cdot\prod_{i'=1}^{i}\prod_{j'=1}^{j}(\alpha + b^2 t_{i'_{j'}})\} \\
\cup \{\forall_{k\in[0,\tau],m\in[1,\psi_k]} \quad & -t_{k_m}, (bID_{k_m} + \nu_w)t_{k_m}, (\alpha + b^2 t_{k_m})\cdot\prod_{k'=1}^{k}\prod_{m'=1}^{\rho'_k}(\alpha + b^2 t_{k'_{m'}})\} \\
\cup \{\forall_{\ell\in[1,q]} \quad & \big(b\cdot\prod_{i=1}^{\tau}\prod_{j=1}^{\rho_i}(\alpha + b^2 t_{i_j})\big)s_\ell, \big(b^2\cdot\prod_{i=1}^{\tau}\prod_{j=1}^{\rho_i}(\alpha + b^2 t_{i_j})\cdot ID_\ell + \nu_w^b\big)s_\ell\}
\end{aligned}
$$

$$Q = \{1\}$$
$$\cup \{\forall_{i \in [0,\tau], j \in [1,\rho_i]} \qquad \alpha \cdot \prod_{i'=1}^{i} \prod_{j'=1}^{j} (\alpha + b^2 t_{i'_{j'}})\}$$

and $f = \alpha s \cdot \prod_{i=1}^{\tau} \prod_{j=1}^{\rho_i} (\alpha + b^2 t_{i_j})$

Boneh et al. show that assumptions of this form are generically secure under few conditions. In particular, let p be the order of the groups, s be the maximum between the number of polynomials in each of P and Q. Let d be twice the maximum degree of any polynomial in (P, Q, f). The advantage of an attack algorithm, that makes at most y queries to the oracles computing the group operation in $\mathbb{G}, \mathbb{G}_T$ and the bilinear pairing $e : \mathbb{G}^2 \to \mathbb{G}_T$, is bounded by

$$Adv(\mathcal{A}) \leq \frac{(y + 2s + 2)^2 \cdot d}{2p}$$

as long as f is symbolically independent of (P, Q), i.e. there is no linear combination of polynomials from Q and multiplications of pairs of polynomials from P that is symbolically equal to f [6].

The maximum degree of any polynomial in the assumption is $3n + 3$ and $s = 2q + 3n + 3(n - q)$. To realize f from (P, Q) we need to have a term of the form $\alpha s \cdot \prod_{i=1}^{\tau} \prod_{j=1}^{\rho_i} (\alpha + b^2 t_{i_j})$. We note that no such terms can be realized from the product of any two polynomials $p, p' \in P$. We let $p = s$, in this case, p' must be of the form $(\alpha + b^2 t_{k_m}) \cdot \prod_{k'=1}^{k} \prod_{m'=1}^{\rho'_k} (\alpha + b^2 t_{k'_{m'}})$. There are two cases:

1. t_{k_m} corresponds to a temporary revoked user. We show that $sb^2 t_{k_m}$ cannot be realized. In order to realize that term we have two cases:
   (a) Use $\big(b^2 \cdot \prod_{i=1}^{\tau} \prod_{j=1}^{\rho_i} (\alpha + b^2 t_{i_j}) \cdot ID_\ell + \nu_w^b\big) s_\ell$
   However, this creates a w^{bs_ℓ} term that can only be canceled by a product of $(bID_{k_m} + \nu_w) t_{k_m}$ and $(b \cdot \prod_{i=1}^{\tau} \prod_{j=1}^{\rho_i} (\alpha + b^2 t_{i_j}) s_\ell)$. In turn, this creates a $b^2 t_{k_m}$ term that can only be canceled by a product of $(-t_{k_m})$ and $\big(b^2 \cdot \prod_{i=1}^{\tau} \prod_{j=1}^{\rho_i} (\alpha + b^2 t_{i_j}) \cdot ID_\ell + \nu_w^b\big) s_\ell$. This leads us to $b^2 s_\ell t_{k_m} (ID_{k_m} - ID_\ell)$. Since t_{k_m} corresponds to a temporary revoked user, there exists an ℓ in $[1, q]$ such that $ID_{k_m} = ID_\ell$ and $b^2 s_\ell t_{k_m}$ cannot be realized. Since $s = \sum s_\ell$, $sb^2 t_{k_m}$ cannot be realized.
   (b) Use $(bID_{k_m} + \nu_w) t_{k_m}$. This case is symmetric to the previous case.
2. t_{k_m} corresponds to a permanent revoked user. We note that the product $\prod_{m'=1}^{\rho'_k} (\alpha + b^2 t_{k'_{m'}})$ cannot be altered to include the term $(\alpha + b^2 t_{k_m})$ which is

[6] We refer to [BBG05] for formal definitions

include in $\prod_{i=1}^{\tau} \prod_{j=1}^{\rho_i} (\alpha + b^2 t_{i_j})$ since t_{k_m} corresponds to a permanent revoked user.

It follows from [BBG05], assuming the adversary makes at most y queries, that

$$Adv^{\mathrm{nqdme}}(n, q, \mathcal{A}) \leq \frac{(y + 2(6n - q) + 2)^2 \cdot (3n + 3)}{2p}$$

When $y > n$ we have that the advantage is $O(y^2 n/p)$.