# Broadcast Encryption with Both Temporary and Permanent Revocation

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

[1] Department of Computer Science, Ben-Gurion University of the Negev, Beersheba, Israel
{danbr,dolev}@cs.bgu.ac.il
[2] Department of Communication Systems Engineering, Ben-Gurion University of the Negev, Beersheba, 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 the same performance as state of the art schemes that achieve only one of the two types of revocation. 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.

S. Dolev—This research was partially supported by the Rita Altura Trust Chair in Computer Sciences; the Lynne and William Frankel Center for Computer Science; grant of the Ministry of Science, Technology and Space, Israel, and the National Science Council (NSC) of Taiwan; the Ministry of Foreign Affairs, Italy; the Ministry of Science, Technology and Space, Infrastructure Research in the Field of Advanced Computing and Cyber Security and the Israel National Cyber Bureau.
N. Gilboa—Supported by ISF grant 1638/15, a grant by the BGU Cyber Center, the Israeli Ministry Of Science and Technology Cyber Program and by the European Union's Horizon 2020 ICT program (Mikelangelo project).


P. Spirakis and P. Tsigas (Eds.): SSS 2017, LNCS 10616, pp. 469–483, 2017.
https://doi.org/10.1007/978-3-319-69084-1_35

## 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[1]. 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 a content 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 encrypted wit this content encryption key and is broadcast to all users in the system, but only the authorized users who received the 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[2] 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 parameters. 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

[1] The work of Delerablée et al. [DPP07] is an exception, considering both types of revocation.

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

are $O(1)$ and revocation is especially trivial for all users actually requiring *less* 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][3]. 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

[3] The first scheme of Delerablée et al. [DPP07] is a public-key construction with public key of size $O(n)$ for $n$ users.

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. The security of the scheme is proved in the generic group model which implies that any attack on the system must rely on the representation of the group used to implement the scheme.

The generic group model was introduced by Shoup in [Sho97] and extended by Boneh et al. in [BBG05] to groups $\mathbb{G}$ with prime order $p$ that are endowed with a bilinear map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$. [BBG05] introduces a General Decisional Diffie-Hellman Exponent assumption, which is in fact a family of hardness assumptions that include many, but not all, hardness assumptions over bilinear groups. This setting defines two sequences $P, Q \in \mathbb{F}_p[x_1, \ldots, x_n]^s$ of multivariate polynomials and an additional polynomial $f \in \mathbb{F}_p[x_1, \ldots, x_n]$. The adversary receives two sequences of elements $(g^{P(x_1,\ldots,x_n)}, e(g,g)^{Q(x_1,\ldots,x_n)}) \in \mathbb{G}^s \times \mathbb{G}_{\mathbb{T}}{}^s$ for a generator $g \in \mathbb{G}$ and tries to distinguish between $e(g,g)^{f(x_1,\ldots,x_n)}$ and a random element in $\mathbb{G}_T$. A theorem in [BBG05] shows that any instance of the General Decisional Diffie-Hellman Exponent problem is secure in the generic group model as long as there doesn't exist a linear combination of quadratic polynomials in $P$ and of $Q$, which is equal to $f$. A different way to view this result is that in the generic group model the adversary is restricted to group operations and bilinear mappings on elements of $\mathbb{G}$ and to group operations on elements of $\mathbb{G}_T$ and if they don't equal $g^{f(x_1,\ldots,x_n)}$ then that element appears random.

The General Decisional Diffie-Hellman Exponent setting does not cover problems in which the adversary is given functions of the secrets $x_1, \ldots, x_n \in \mathbb{F}_p$ in addition to $(g^{P(x_1,\ldots,x_n)}, e(g,g)^{Q(x_1,\ldots,x_n)})$. Such is the case for the construction in [DP08].

A second contribution of our work consists of defining the Diffie-Hellman Mixed Exponent Assumption (DH-MEA) which generalizes the General Decisional Diffie-Hellman Exponent by adding functions of the exponents $x_1, \ldots, x_n$ to the information that adversary receives. The DH-MEA is a family of assumptions in which a specific member is defined by three sequences of multivariate polynomials $P, Q, Z \in \mathbb{F}_p[x_1, \ldots, x_n]^s$ and an additional polynomial $f \in \mathbb{Z}[x_1, \ldots, x_n]$. The adversary receives the pair $(g^{P(x_1,\ldots,x_n)}, e(g,g)^{Q(x_1,\ldots,x_n)})$ and $Z(x_1, \ldots, X_n)$ and must distinguish between $e(g,g)^{f(x_1,\ldots,x_n)}$ and a random element in $\mathbb{G}_T$.

While in the generic group model the adversary is limited in the way it can manipulate the group elements $g^{P(x_1,\ldots,x_n)}$ and $e(g,g)^{Q(x_1,\ldots,x_n)}$, there is no such limitation when it is presented with a function $z(x_1, \ldots, x_n) \in \mathbb{F}_p$. If there exists a linear combination of polynomials of two types: $\nu_{i,j}(Z(x_1, \ldots, x_n))p_i p_j$ and $\mu_k(Z(x_1, \ldots, x_n))q_k$ that is equal to $f$ when $p_i, p_j$ are part of $P$, $q_k$ is part of

$Q$ and $\nu_{i,j}, \mu_k$ are arbitrary functions over $\mathbb{F}_p$ then the adversary can break the assumption since it can test whether the challenge is $e(g,g)^{f(x_1,\ldots,x_n)}$. We show that if such a combination *does not* exist then the DH-MEA assumption is secure in the generic group model.

We prove the security of our broadcast encryption scheme by showing that what an adversary learns in the security game is an instance of the DH-MEA. Security of the broadcast encryption scheme in the generic group model follows from the general theorem on DH-MEA.

Our construction has similar performance to a combination of the performance of [DPP07,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 $\lambda$ 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 $\mathsf{false}$.

$\mathsf{Revoke}(S, PP, MSK)$. The revocation algorithm takes as input the master secret key $MSK$, the public parameters $PP$ and a set $S$ of identities to (permanently) 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 (temporarily) 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 $i = 1, \ldots, n$:

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

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

**Security.** The security of a scheme with both permanent and temporary revocation 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 $\lambda$ 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 revocation scheme with security parameter $\lambda$ is defined as

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

A scheme with both permanent and temporary revocation 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 $\mathsf{Init}$ phase.

### 2.2 Bilinear Maps

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

1. Bilinearity. For every $g_1, g_2 \in \mathbb{G}$ and $\alpha \in \mathbb{F}_p$ 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 Decision Diffie-Hellman Mixed Exponent Problem

**Notation 1.** *For a prime $p$ and field with $p$ elements, $\mathbb{F}_p$, let $\mathbb{F}_p[X]$ denote the ring of polynomials in $n$ variables $X = x_1, \ldots, x_n$ over $\mathbb{F}_p$. Let $Z, P, Q \in \mathbb{F}_p[X]^s$ be three sequences of $s$ polynomials, which we denote by $P = (p_1, \ldots, p_s), Q = (q_1, \ldots, q_s), Z = (z_1, \ldots, z_s)$ and let $p_1 = q_1 = 1$. Let $f \in \mathbb{F}_p[X]$ be the* target *polynomial.*

Let $\mathbb{G}$ be a bilinear group of order $p$ with target group $\mathbb{G}_T$, let $g$ be a generator of $\mathbb{G}$ and let $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_t$ be a bilinear mapping. The decision Diffie-Hellman Mixed Exponent problem is defined as follows.

**Definition 1.** *Let $H(X) = (Z(X), g^{P(X)}, e(g,g)^{Q(X)}) \in \mathbb{Z}_p^s \times \mathbb{G}^s \times \mathbb{G}_t^s$. We say that an algorithm $\mathcal{B}$ has advantage $\epsilon$ in the Decision $(Z, P, Q, f)$-Diffie-Hellman mixed exponent problem in $\mathbb{G}$ if*

$$\left| Pr[\mathcal{B}(H(X), e(g,g)^{f(X)}) = 0] - Pr[\mathcal{B}(H(X), T) = 0] \right| > \epsilon$$

*where $T \in \mathbb{G}_t$ is chosen uniformly at random and the probability is taken over the random choices of $g, X, T$ and the random bits consumed by $\mathcal{B}$.*

Intuitively, for some combinations of polynomial sequences $Z, P, Q$ and $f$ this decision problem is easy. The following definition addresses such combinations:

**Definition 2.** *Let $Z, P, Q \in \mathbb{F}_p[X]^s$, where $p_1 = q_1 = 1$ and let $f \in \mathbb{F}_p[X]$. We say that $f$ is dependent on $(Z, P, Q)$ if there exist functions $\{\nu_{i,j}\}_{i,j=1}^s, \{\mu_k\}_{k=1}^s : \mathbb{Z}_p^s \to \mathbb{Z}_p$ such that*

$$f = \sum_{i,j=1}^{s} \nu_{i,j}(Z(X_1, \ldots, X_n)) p_i p_j + \sum_{k=1}^{s} \mu_k(Z(X_1, \ldots, X_n)) q_k$$

*We say that $f$ is independent of $Z, P$ and $Q$ if it is not dependent on them.*

## 3 Public Key Revocation Scheme

Setup $(\lambda)$. The setup algorithm, given a security parameter $\lambda$, 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 $\phi$[4] from $F_\lambda$, 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

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

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 $SK_{ID} = \{D_1, \ldots, D_5\}$.

*Revoke* $(S, PP, MSK)$. The algorithm is given a set $S = \{ID_1, \ldots, ID_r\}$ 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 $g^{bST}, g^{b^2ST}, w^{bST}$ and $e(g,g)^{\alpha ST}$ with $g^{bST'}, g^{b^2ST'}, w^{bST'}$ 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 $h_i = \left(\frac{S_{i,2}}{h_{i-1}}\right)^{\frac{1}{D_3 - S_{i,1}}}$. Finally, the algorithm updates $SK_{ID}$ by replacing $D_4$ with $h_r$.

[4] We slightly abuse notation and use $\phi$ to denote both the function and a concrete description of this function.

We note that $h_r = g^{(\alpha+b^2t)\cdot ST}$ where $ST$ is the new state in the master secret key after the corresponding revocation. For example, if $ST = 1$, $t = \phi(ID)$ and $\hat{t} = \phi(\hat{ID})$, then the update process of $SK_{ID}$ after the revocation of $\hat{ID}$ is

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

*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 $r$ revoked identities. The algorithm randomly chooses $s_1, \ldots, s_r \in \mathbb{Z}_p$, computes $s = \sum_{i=1}^{r} s_i$, sets

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

and for $i = 1$ to $r$ it sets

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

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 $D_5 = \text{true}$ or $ID \in \bigcup_{i=1}^{r} C_{i,1}$ the algorithm outputs $\perp$. Otherwise the algorithm calculates:

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

Finally the algorithm retrieves the message

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

## 4 Security Analysis

We prove the security of our construction in the generic group model in three stages. We first state a theorem that the DH-MEA problem is hard in the generic group model. We then show how to transform an attack on the broadcast encryption system to an attack on an ad hoc security assumption that we refer to as the $n - q$ Decisional Assumption ($n - q$ DA). Finally, we prove that the $n - q$ DA is an instance of DH-MEA and is therefore generically secure.

### 4.1 Generic Security of DH-MEA

Recall that the DH-MEA is easy when $f$ is dependent on $(Z, P, Q)$. While it is possible that for some *specific* groups the problem is easy even when $f$ is independent of $(Z, P, Q)$, the following result shows that the independence of $f$ implies security in the generic group model in which group operations and bilinear mappings are provided by oracles.

**Theorem 1.** *Let* $Z = (z_1, \ldots, z_s), P = (p_1, \ldots, p_s), Q = (q_1, \ldots, q_s) \in \mathbb{F}_p[X]^s$, $p_1 = q_1 = 1$ *and let* $f \in \mathbb{F}_p[X_1, \ldots, X_n]$. *If* $f$ *is independent of* $(Z, P, Q)$ *and* $deg = \max\{2deg_P, deg_f, deg_Q\}$ *then the advantage of any generic adversary* $\mathcal{A}$ *that performs at most* $y$ *queries to the oracles (for group operations in* $\mathbb{G}, \mathbb{G}_T$ *and evaluations of* $e$*) in the Decision* $(Z, P, Q, f)$*-Diffie-Hellman Mixed Exponent Problem is bounded by:*

$$Adv(\mathcal{A}) = O(\frac{(y+s)^2 \cdot deg}{p})$$

The full proof is omitted due to space constraints and will appear in the full version of the paper.

**Corollary 1.** *For* $Z, P, Q$ *and* $f$ *as in Theorem 1, if* $f$ *is independent of* $(Z, P, Q)$ *and* $deg = \max\{2deg_P, deg_f, deg_Q\}$ *then any adversary* $\mathcal{A}$ *that has advantage* $1/2$ *in solving the decision* $(Z, P, Q, f)$*-Diffie-Hellman mixed exponent problem in a generic bilinear group* $\mathbb{G}$ *must make at least* $\Omega(\sqrt{p/deg} - s)$ *queries to the group oracles.*

### 4.2 Security of the Broadcast Encryption System

**Theorem 2.** *The scheme in Sect. 3 is a broadcast encryption system with permanent and temporary revocation which is adaptively secure in the generic group model.*

*Proof.* We first write the elements that an adversary learns during the security game, from which we state a computational assumption. Let $\tau$ be the number of permanent revocation requests that the adversary performs. Let $\rho_i$ 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 $\psi_i$ denote the number of secret key requests the adversary performs after the $i$-th permanent revocation request ($\psi_0$ 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 $\phi(ID_{k_m})$. Let $q$ denote the number of users the adversary revoke during the temporary revocation. We denote their identities by $ID_i$ where $i$ in $[1, q]$.

From the public parameters and revocation requests, the adversary learns

$$\forall i \in [0, \tau], j \in [1, \rho_i]\ g^{ST_{i,j}}, g^{b \cdot ST_{i,j}}, g^{b^2 \cdot ST_{i,j}}, w^{b \cdot ST_{i,j}}, e(g, g)^{\alpha \cdot ST_{i,j}}$$

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

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

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

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

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

The adversary obtains keys only for identities $ID_{k_m}$ such 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.

**The $(n-q)$-Decisional 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

$$\sum_{k=0}^{\tau} \psi_k = n \bmod p \text{ and } \sum_{i=1}^{\tau} \rho_i = n - q \bmod p$$

the $(n-q)$-Decisional problem is defined as follows. A challenger chooses generators $g, w \in \mathbb{G}$ and random exponents $\alpha, b, \gamma, \{t_{k_m}\}_{k \in [0,\tau], m \in [1,\psi_k]} \in \mathbb{Z}_p$. Suppose an adversary is given $\mathbf{X} =$

$$\forall_{i \in [0,\tau], j \in [1,\rho_i]} \quad \left\{ \frac{1}{\prod_{i'=1}^{i} \prod_{j'=1}^{j} (\alpha + b^2 t_{i'_{j'}})} - \gamma, g^{\prod_{i'=1}^{i} \prod_{j'=1}^{j} (\alpha + b^2 t_{i'_{j'}})}, \right.$$
$$\left. g^{b \cdot \prod_{i'=1}^{i} \prod_{j'=1}^{j} (\alpha + b^2 t_{i'_{j'}})}, g^{b^2 \cdot \prod_{i'=1}^{i} \prod_{j'=1}^{j} (\alpha + b^2 t_{i'_{j'}})}, \right.$$
$$\left. w^{b \cdot \prod_{i'=1}^{i} \prod_{j'=1}^{j} (\alpha + b^2 t_{i'_{j'}})}, e(g,g)^{\alpha \cdot \prod_{i'=1}^{i} \prod_{j'=1}^{j} (\alpha + b^2 t_{i'_{j'}})} \right.$$

$$\forall_{k \in [0,\tau], m \in [1,\psi_k]} \quad \left\{ ID_{k_m}, g^{-t_{k_m}}, (g^{bID_{k_m}} w)^{t_{k_m}}, \frac{1}{\alpha + b^2 t_{k_m}} - \gamma, \right.$$
$$\left. g^{(\alpha + b^2 t_{k_m}) \cdot \prod_{k'=1}^{k} \prod_{m'=1}^{\rho'_k} (\alpha + b^2 t_{k'_{m'}})} \right.$$

$$g^s$$

$$\forall_{\ell \in [1,q]} \quad (g^{b \cdot \prod_{i=1}^{\tau} \prod_{j=1}^{\rho_i} (\alpha + b^2 t_{i_j})})^{s_\ell}, (g^{b^2 \cdot \prod_{i=1}^{\tau} \prod_{j=1}^{\rho_i} (\alpha + b^2 t_{i_j}) \cdot ID_\ell} w^b)^{s_\ell}$$

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

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

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

$$Adv^{\mathrm{nqd}}(n,q,\mathcal{A}) := |Pr[\mathcal{A}(\mathbf{X},T)] - Pr[\mathcal{A}(\mathbf{X},R)]| \geq \epsilon$$

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

It is clear that the $(n-q)$ DA is equivalent to breaking the broadcast encryption scheme. However, showing that it is an instance of the DH-MEA requires to present it using the terminology of Definition 1 as a $(Z,P,Q,f)$ mixed exponent problem (denoting $w = g^\omega$).

$$Z = \{\forall_{\substack{i\in[0,\tau]\\ j\in[1,\rho_i]}} \quad \frac{1}{\prod_{i'=1}^{i}\prod_{j'=1}^{j}(\alpha+b^2 t_{i'_{j'}})} - \gamma\}$$

$$P = \{1, s\}$$

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

$$\cup \{\forall_{\substack{k\in[0,\tau]\\ m\in[1,\psi_k]}} \quad -t_{k_m}, (bID_{k_m}+\omega)t_{k_m}, (\alpha+b^2 t_{k_m})\cdot\prod_{k'=1}^{k}\prod_{m'=1}^{\rho'_k}(\alpha+b^2 t_{k'_{m'}})\}$$

$$\cup \{\forall_{\ell\in[1,q]} \quad \left(b\cdot\prod_{i=1}^{\tau}\prod_{j=1}^{\rho_i}(\alpha+b^2 t_{i_j})\right)s_\ell, \left(b^2\cdot\prod_{i=1}^{\tau}\prod_{j=1}^{\rho_i}(\alpha+b^2 t_{i_j})\cdot ID_\ell+\omega^b\right)s_\ell\}$$

$$Q = \{1\}$$

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

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

The maximum degree of $f$ and of any polynomial in $P, Q$ is $3n+3$ and the number of polynomials in each of $P$ and $Q$ is at most $2q+3n+3(n-q)$.

Therefore, by Corollary 1 if we prove that $f$ is independent of $(Z, P, Q)$ we are done since to have a noticeable advantage in the security game the adversary must make an exponential number of oracle queries.

Since $f = \alpha s \cdot \prod_{i=1}^{\tau} \prod_{j=1}^{\rho_i} (\alpha + b^2 t_{i_j})$ is a product of terms including $s$ and $s$ appears in a single polynomial in $Z, P$ or $Q$ that polynomial, which is $s$ itself, must be part of any combination of elements that is equal to $f$. Any function of a single element in $Z$ is not equal to $\prod_{j=1}^{\rho_i} (\alpha + b^2 t_{i_j})$ due to the masking by $\gamma$. A function of two elements or more from $Z$ can remove $\gamma$ but at the cost of creating sums of elements in $Z$ such that again any function on them is not equal to $\prod_{j=1}^{\rho_i} (\alpha + b^2 t_{i_j})$.

Therefore, producing $\prod_{j=1}^{\rho_i} (\alpha + b^2 t_{i_j})$ must use a linear combination of elements of $P$ which will then be multiplied with $s$. Note that the coefficients of the polynomials of $P$ can be arbitrary functions of $Z$. The only useful polynomials in $P$ for this purpose are 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 temporarily 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 $\left(b^2 \cdot \prod_{i=1}^{\tau} \prod_{j=1}^{\rho_i} (\alpha + b^2 t_{i_j}) \cdot ID_\ell + \omega^b\right) s_\ell$
   However, this creates a $w^{bs_\ell}$ term that can only be canceled by a product of $(bID_{k_m} + \omega) t_{k_m}$ 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 $\left(b^2 \cdot \prod_{i=1}^{\tau} \prod_{j=1}^{\rho_i} (\alpha + b^2 t_{i_j}) \cdot ID_\ell + \omega^b\right) s_\ell$. This leads us to $b^2 s_\ell t_{k_m} (ID_{k_m} - ID_\ell)$. Since $t_{k_m}$ corresponds to a temporarily revoked user, there exists an $\ell$ 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 $(bID_{k_m} + \omega) t_{k_m}$. This case is symmetric to the previous case.
2. $t_{k_m}$ corresponds to a permanently 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 part of $\prod_{i=1}^{\tau} \prod_{j=1}^{\rho_i} (\alpha + b^2 t_{i_j})$ since $t_{k_m}$ corresponds to a permanently revoked user. To see why that is the case, it might be easier to denote $\frac{1}{(\alpha + b^2 t_{i_j})} - \gamma$ by $x_{ij}$. In this representation, the task is to calculate $\frac{1}{(x_{ij} - \gamma)^2}$ from the pair $(x_{ij}, \frac{1}{(x_{ij} - \gamma)})$. Recall that $x_{ij} \in Z$, $\frac{1}{(x_{ij} - \gamma)} \in P$ and since it is only possible to do additions of elements in $P$, knowing $x_{ij}$ is of no value.

It follows from Corollary 1, that in order to break the assumption with non-negligible probability, the adversary must make at least $O(\sqrt{p/n})$ queries.

## References


[BBG05] Boneh, D., Boyen, X., Goh, E.-J.: Hierarchical identity based encryption with constant size ciphertext. IACR Cryptology ePrint Archive 2005:15 (2005)

[BGW05] Boneh, D., Gentry, C., Waters, B.: Collusion resistant broadcast encryption with short ciphertexts and private keys. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 258–275. Springer, Heidelberg (2005). doi:10.1007/11535218_16

[CGI+99] Canetti, R., Garay, J.A., Itkis, G., Micciancio, D., Naor, M., Pinkas, B.: Multicast security: a taxonomy and some efficient constructions. In: INFOCOM, pp. 708–716. IEEE (1999)

[CMN99] Canetti, R., Malkin, T., Nissim, K.: Efficient communication-storage tradeoffs for multicast encryption. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 459–474. Springer, Heidelberg (1999). doi:10.1007/3-540-48910-X_32

[DF02] Dodis, Y., Fazio, N.: Public key broadcast encryption for stateless receivers. In: Feigenbaum, J. (ed.) DRM 2002. LNCS, vol. 2696, pp. 61–80. Springer, Heidelberg (2003). doi:10.1007/978-3-540-44993-5_5

[DP08] Delerablée, C., Pointcheval, D.: Dynamic threshold public-key encryption. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 317–334. Springer, Heidelberg (2008). doi:10.1007/978-3-540-85174-5_18

[DPP07] Delerablée, C., Paillier, P., Pointcheval, D.: Fully collusion secure dynamic broadcast encryption with constant-size ciphertexts or decryption keys. In: Takagi, T., Okamoto, E., Okamoto, T., Okamoto, T. (eds.) Pairing 2007. LNCS, vol. 4575, pp. 39–59. Springer, Heidelberg (2007). doi:10.1007/978-3-540-73489-5_4

[FN93] Fiat, A., Naor, M.: Broadcast encryption. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 480–491. Springer, Heidelberg (1994). doi:10.1007/3-540-48329-2_40

[GST04] Goodrich, M.T., Sun, J.Z., Tamassia, R.: Efficient tree-based revocation in groups of low-state devices. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 511–527. Springer, Heidelberg (2004). doi:10.1007/978-3-540-28628-8_31

[GSW00] Garay, J.A., Staddon, J., Wool, A.: Long-lived broadcast encryption. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 333–352. Springer, Heidelberg (2000). doi:10.1007/3-540-44598-6_21

[GW09] Gentry, C., Waters, B.: Adaptive security in broadcast encryption systems (with short ciphertexts). In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 171–188. Springer, Heidelberg (2009). doi:10.1007/978-3-642-01001-9_10

[LSW10] Lewko, A.B., Sahai, A., Waters, B.: Revocation systems with very small private keys. In: IEEE Symposium on Security and Privacy, pp. 273–285. IEEE Computer Society (2010)

[NNL01] Naor, D., Naor, M., Lotspiech, J.: Revocation and tracing schemes for stateless receivers. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 41–62. Springer, Heidelberg (2001). doi:10.1007/3-540-44647-8_3

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

[Sho97] Shoup, V.: Lower bounds for discrete logarithms and related problems. In: Fumy, W. (ed.) EUROCRYPT 1997. LNCS, vol. 1233, pp. 256–266. Springer, Heidelberg (1997). doi:10.1007/3-540-69053-0_18