Identity-Based Broadcast Encryption with Constant Size Ciphertexts and Private Keys

Cécile Delerablée[1,2]

[1] Orange Labs - Caen, France
[2] ENS - Paris, France
cecile.delerablee@orange-ftgroup.com

Abstract. This paper describes the first identity-based broadcast encryption scheme (IBBE) with constant size ciphertexts and private keys. In our scheme, the public key is of size linear in the maximal size m of the set of receivers, which is smaller than the number of possible users (identities) in the system. Compared with a recent broadcast encryption system introduced by Boneh, Gentry and Waters (BGW), our system has comparable properties, but with a better efficiency: the public key is shorter than in BGW. Moreover, the total number of possible users in the system does not have to be fixed in the setup.

1 Introduction

Broadcast Encryption. The concept of Broadcast Encryption (BE) was introduced by Fiat and Naor in [16]. In BE schemes, a broadcaster encrypts messages and transmits them to a group of users who are listening to a broadcast channel and use their private keys to decrypt transmissions. At encryption time, the broadcaster can choose the set $\mathcal{S}$ of identities that will be able to decrypt messages. A BE scheme is said to be fully collusion resistant when, even if all users that are not in $\mathcal{S}$ collude, they can by no means infer information about the broadcast message.

Many BE systems have been proposed [23,20,19,10,15]. The best known fully collusion systems are the schemes of Boneh, Gentry and Waters [10] which achieve $O(\sqrt{n})$-size ciphertexts and public key, or constant size ciphertexts, $O(n)$-size public key and constant size private keys in a construction that we denote by BGW_1 in the following. A lot of systems make use of the hybrid (KEM-DEM) encryption paradigm where the broadcast ciphertext only encrypts a symmetric key used to encrypt the broadcast contents. We will adopt this methodology in the following.

Dynamic Broadcast Encryption. The concept of Dynamic Broadcast Encryption (DBE) was introduced by Delerablée, Paillier and Pointcheval in [15]. A DBE scheme is a BE in which the total number of users is not fixed in the setup, with the property that any new user can decrypt all previously distributed messages. Thus a DBE scheme is suitable for some applications, like DVD encryption.

K. Kurosawa (Ed.): ASIACRYPT 2007, LNCS 4833, pp. 200–215, 2007.

Nevertheless, some applications like Video on Demand (VOD) need forward secrecy. This paper address this problem, in the identity-based setting.

ID-based Encryption. In 1984, Shamir [24] asked for a public key encryption scheme in which the public key can be an arbitrary string.

Since the problem was posed in 1984, there have been several proposals for Identity-Based Encryption (IBE) schemes. However, we can considerer that the first practical IBE scheme was introduced by Boneh and Franklin in 2001 [9]. Since 2001, several schemes have been introduced [14,26,12,8,7,6,17]. Concerning the security, there are mainly two definitions:

1. Full security, which means that the attacker can choose adaptively the identity he wants to attack (after having seen the parameters);
2. Selective-ID security, which means that the attacker must choose the identity he wants to attack at the beginning, before seeing the parameters. The Selective-ID security is thus weaker than full security.

Since the scheme in [9] is proved secure in the random oracle model, several papers have proposed systems secure without random oracles. In [6], one of the systems has short parameters and tight security reduction, in the standard model (proved secure against selective-ID adversaries). In [17], Gentry proposed the first IBE system that is fully secure without random oracles, has short public parameters and has a tight security reduction.

Multi-receiver ID-based Key Encapsulation (mID-KEM). A multi-receiver key encapsulation scheme (mKEM) is an efficient key encapsulation mechanism for multiple parties. This notion was introduced in [25]. Note that this notion is different from multi-recipient public key encryption [4,5,22], where the sender wants to send one (different) message to each receiver.

Later, in [2] and [3], the notion of mKEM was extended to multi-receiver identity-based key encapsulation (mID-KEM), i.e. mKEM in the identity-based setting. In [2] and [3], the ciphertext size grows with the number of receivers. In [13], Chatterjee and Sarkar achieved a controllable trade-off between the ciphertext size and the private key size: ciphertexts are of size $|\mathcal{S}|/N$, and private keys are of size N where $\mathcal{S}$ is the set of receivers and N a parameter of the protocol (which also represents, in the security reduction, the maximum number of identities that the adversary is allowed to target). Thus they introduced the first mID-KEM protocols to achieve sub-linear ciphertext sizes. Very recently, Abdalla et al. proposed in [1] a generic construction that achieves ciphertexts of constant size, but private keys of size $O({n_{max}}^2)$.

In the following, we do not employ the term "mID-KEM" anymore, but we talk about "identity-based broadcast encryption" (IBBE), to emphasize that this notion is close to broadcast encryption and ID-based encryption. We consider IBBE as a natural generalization of IBE. Indeed, in IBE schemes, one public key can be used to encrypt a message to any possible identity. In an IBBE scheme, one public key can be used to encrypt a message to any possible group of s identities. Consequently, if we set $s = 1$, the resulting IBBE scheme is an IBE scheme. The trivial solution to construct an IBBE scheme would be to use an IBE scheme to encrypt the message once for each identity. The resulting ciphertext would be of size linear in s. We also see IBBE as a way to make broadcast encryption more practical.

Motivations. We focus on schemes with ciphertexts of constant size. In BGW_1, as we said before, the public key is linear in the total number of decryption keys that can be distributed. Moreover, this number is fixed in the setup. Thus one of our motivations is to introduce a system in which the number of possible decryption keys is not fixed in the setup, and thus does not have any impact on the size of the public key. In [13] and [1], the trade-off between the ciphertext size and the private key size implies that if we want to have short ciphertexts, the private keys cannot be of constant size. Thus we would like to have both ciphertexts and private keys of constant size (as in BGW_1). Note that in some systems like the HIBE scheme in [8], the size of the public key can be reduced by using a hash function, viewed as a random oracle in the security proof, but this is not the case in BGW_1, because all the elements of the public depend on a single value.

Our contributions. In this paper, we propose the first identity-based broadcast encryption scheme with constant size ciphertexts *and* private keys. Our construction is a Key Encapsulation Mechanism (KEM), thus long messages can be encrypted under a short symmetric key. In our solution, ciphertexts and private keys are of constant size, and the public key is linear in the maximal value of s. Moreover, in our scheme, the Private Key Generator ($\mathcal{PKG}$) can dynamically add new members without altering previously distributed information (as in IBE schemes). We also note that there is no hierarchy between identities, contrary to HIBE (Hierarchical IBE [21,18,8]). No organization of the users is needed to have short ciphertexts. Note that the public key is linear in the maximal size of $\mathcal{S}$, and not in the number of decryption keys that can be distributed, which is the number of possible identities. The following framework is an example to show the benefits of our solution: The $\mathcal{PKG}$ can send short term decryption keys. Then sending a new decryption key could be conditional (each month, if the user pays his bill for example), without affecting the performances of the system. Indeed, there is no need to revoke previous keys, because the encryption takes into account the set of users who can decrypt. We can compare our scheme with BGW_1 in such a situation: if we consider that the number of users who can decrypt is s, and that each user receives a new key at the end of each time period, then the size of the public key in BGW_1 would be $\lambda_{\mathsf{PK}} = s \cdot t$ with t the number of time periods for example. In our scheme, we have $\lambda_{\mathsf{PK}} = s$. Thus one can note that BGW_1 is not really suited to such an situation (the public key would grow linearly with the number of time periods). In other words, in BGW_1, the public key is linear in the number of private keys that can be distributed, whereas in our construction, the public key is linear in the maximal number of receivers of a ciphertext, which is independent of the number of private keys that can be distributed. Indeed, in our case, the number of possible private keys is the number of possible identities. Note that if there are n receivers and it happens that $n > m$, we can just concatenate several encryptions together and get n/m size ciphertexts (as in [13]), still with constant size private keys. Moreover, in our construction, ciphertext size is deterministic whereas [13] makes probabilistic efficiency claims.

2 Preliminaries

We propose a formal definition of an identity-based broadcast encryption scheme and security notions that we associate to it. We basically include an $\mathsf{Extract}$ procedure in the definition of Broadcast Encryption given in [10]. Our formal model can also be viewed as a generalization of classical IBE systems. Concerning the security, we follow the definition of the classical security notions for BE (security against static adversaries) [10], which is close to the notion of selective-ID security, used in [6,11].

2.1 Identity-Based Broadcast Encryption (IBBE)

An IBBE scheme involves an authority: the Private Key Generator ($\mathcal{PKG}$). The $\mathcal{PKG}$ grants new members capability of decrypting messages by providing each new member (with identity ID_i) a decryption key $\mathsf{sk}_{\mathsf{ID}_i}$. The generation of $\mathsf{sk}_{\mathsf{ID}_i}$ is performed using a master secret key MSK. The broadcaster encrypts messages and transmits these to the group of users via the broadcast channel. In a (public-key) IBBE encryption scheme, the broadcaster does not hold any private information and encryption is performed with the help of a public key PK and identities of the receivers. Following the KEM-DEM methodology, broadcast encryption is viewed as the combination of a specific key encapsulation mechanism (a Broadcast-KEM) with a symmetric encryption (DEM) that shall remain implicit throughout the paper. More formally, an identity-based broadcast encryption scheme $\mathcal{IBBE}$ with security parameter λ and maximal size m of the target set, is a tuple of algorithms $\mathcal{IBBE} = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{Encrypt}, \mathsf{Decrypt})$ described as follows:

$\mathsf{Setup}(\lambda, m)$. Takes as input the security parameter λ and m the maximal size of the set of receivers for one encryption, and outputs a master secret key MSK and a public key PK. The $\mathcal{PKG}$ is given MSK, and PK is made public.

$\mathsf{Extract}(\mathsf{MSK}, \mathsf{ID}_i)$. Takes as input the master secret key MSK and a user identity ID_i. $\mathsf{Extract}$ generates a user private key $\mathsf{sk}_{\mathsf{ID}_i}$.

$\mathsf{Encrypt}(\mathcal{S}, \mathsf{PK})$. Takes as input the public key PK and a set of included identities $\mathcal{S} = \{\mathsf{ID}_1, \ldots, \mathsf{ID}_s\}$ with $s \leq m$, and outputs a pair (Hdr, K), where Hdr is called the header and $K \in \mathcal{K}$ and $\mathcal{K}$ is the set of keys for the symmetric encryption scheme.

When a message $M \in \{0,1\}^*$ is to be broadcast to users in $\mathcal{S}$, the broadcaster generates $(\mathsf{Hdr}, K) \leftarrow \mathsf{Encrypt}(\mathcal{S}, \mathsf{PK})$, computes the encryption C_M of M under the symmetric key $K \in \mathcal{K}$ and broadcasts $(\mathsf{Hdr}, \mathcal{S}, C_M)$. We will refer to Hdr as the header or broadcast ciphertext, $(\mathsf{Hdr}, \mathcal{S})$ as the full header, K as the message encryption key and C_M as the broadcast body.

$\mathsf{Decrypt}(\mathcal{S}, \mathsf{ID}, \mathsf{sk}_{\mathsf{ID}}, \mathsf{Hdr}, \mathsf{PK})$. Takes as input a subset $\mathcal{S} = \{\mathsf{ID}_1, \ldots, \mathsf{ID}_s\}$ (with $s \leq m$), an identity ID and the corresponding private key $\mathsf{sk}_{\mathsf{ID}}$, a header Hdr, and the public key PK. If $\mathsf{ID} \in \mathcal{S}$, the algorithm outputs the message encryption key K which is then used to decrypt the broadcast body C_M and recover M.

Remark. This model defines, when $m = 1$, an IBE system.

2.2 Security Notions for IBBE

The standard notion for BE schemes is Chosen Ciphertext Security against Static Adversaries. For IBE, one standard notion is selective-ID security (weaker than full security), where the adversary must choose at the beginning of the game the set of identities he wants to attack.

Remark. Note that for $m = 1$ the following security model fits with IND-sID-CCA security for IBE schemes, that is used in [6] for example.

IND-sID-CCA Security. We define IND-sID-CCA security of an IBBE system. Security is defined using the following game between an adversary $\mathcal{A}$ and a challenger. We basically refine the definition of [10], by adding extraction queries. Both the adversary and the challenger are given as input m, the maximal size of a set of receivers $\mathcal{S}$.

Init: The adversary $\mathcal{A}$ first outputs a set $\mathcal{S}^* = \{\mathsf{ID}_1^*, \ldots, \mathsf{ID}_s^*\}$ of identities that he wants to attack (with $s \leq m$).

Setup: The challenger runs $\mathsf{Setup}(\lambda, m)$ to obtain a public key PK. He gives $\mathcal{A}$ the public key PK.

Query phase 1: The adversary $\mathcal{A}$ adaptively issues queries $q_1, \ldots, q_{s_0}$, where q_i is one of the following:

- Extraction query (ID_i) with the constraint that $\mathsf{ID}_i \notin \mathcal{S}^*$: The challenger runs $\mathsf{Extract}$ on ID_i and forwards the resulting private key to the adversary.
- Decryption query, which consists of a triple $(\mathsf{ID}_i, \mathcal{S}, \mathsf{Hdr})$ with $\mathcal{S} \subseteq \mathcal{S}^*$ and $\mathsf{ID}_i \in \mathcal{S}$. The challenger responds with $\mathsf{Decrypt}(\mathcal{S}, \mathsf{ID}_i, \mathsf{sk}_{\mathsf{ID}_i}, \mathsf{Hdr}, \mathsf{PK})$.

Challenge: When $\mathcal{A}$ decides that phase 1 is over, the challenger runs $\mathsf{Encrypt}$ algorithm to obtain $(\mathsf{Hdr}^*, K) = \mathsf{Encrypt}(\mathcal{S}^*, \mathsf{PK})$ where $K \in \mathcal{K}$. The challenger then randomly selects $b \leftarrow \{0,1\}$, sets $K_b = K$, and sets K_{1-b} to a random value in $\mathcal{K}$. The challenger returns $(\mathsf{Hdr}^*, K_0, K_1)$ to $\mathcal{A}$.

Query phase 2: The adversary continues to issue queries $q_{s_0+1}, \ldots, q_s$ where q_i is one of the following:

- Extraction query (ID_i), as in phase 1.
- Decryption query, as in phase 1, but with the constraint that $\mathsf{Hdr} \neq \mathsf{Hdr}^*$. The challenger responds as in phase 1.

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

We denote by q_D the total number of Decryption queries and by t the total number of extraction queries that can be issued by the adversary during the game. Viewing t, m, q_D as attack parameters, we denote by $\mathsf{Adv}^{\mathsf{ind}}_{\mathcal{IBBE}}(t, m, q_D, \mathcal{A})$ the advantage of $\mathcal{A}$ in winning the game:

$$\mathsf{Adv}^{\mathsf{ind}}_{\mathcal{IBBE}}(t, m, q_D, \mathcal{A}) = |2 \times \Pr[b' = b] - 1| = |\Pr[b' = 1|b = 1] - \Pr[b' = 1|b = 0]|$$

where the probability is taken over the random coins of $\mathcal{A}$, the challenger and all probabilistic algorithms run by the challenger.

Definition 1. *Let* $\mathsf{Adv}^{\mathsf{ind}}_{\mathcal{IBBE}}(t, m, q_D) = \max_{\mathcal{A}} \mathsf{Adv}^{\mathsf{ind}}_{\mathcal{IBBE}}(t, m, q_D, \mathcal{A})$ *where the maximum is taken over all probabilistic algorithms* $\mathcal{A}$ *running in time* $\mathsf{poly}(\lambda)$*. An identity-based broadcast encryption scheme* $\mathcal{IBBE}$ *is said to be* (t, m, q_D)*-IND-sID-CCA secure if* $\mathsf{Adv}^{\mathsf{ind}}_{\mathcal{IBBE}}(t, m, q_D) = \mathsf{negl}(\lambda)$.

IND-sID-CPA. Analogously to [10], we define semantic security for an IBBE scheme by preventing the attacker from issuing decryption queries.

Definition 2. *We say that an identity-based broadcast encryption system is* (t, m)*-IND-sID-CPA secure if it is* $(t, m, 0)$*-IND-sID-CCA secure.*

Remark. In [10], the choice of $\mathcal{S}^*$ implies a choice of corrupted users, because the total number of users is fixed in the setup. In the model we described before, the corrupted users are not chosen at the beginning but adaptively. We describe below a modification of our model which does not allow adaptive corruptions, as in [10].

Definition 3. (t, m, q_D)*-IND-na-sID-CCA security (non adaptive sID): at initialization time, the attacker outputs a set* $\mathcal{S}^* = \{\mathit{ID}_1^*, \ldots, \mathit{ID}_s^*\}$ *of identities that he wants to attack, and a set* $\mathcal{C} = \{\bar{\mathit{ID}}_1, \ldots, \bar{\mathit{ID}}_t\}$ *of identities that he wants to corrupt (i.e. to obtain the corresponding private key). Thus the attacker issues t extraction queries only at the beginning of the game.*

Definition 4. *We say that an identity-based broadcast encryption system is* (t, m)*-IND-na-sID-CPA secure if it is* $(t, m, 0)$*-IND-na-sID-CCA secure.*

Full collusion resistance. In an IBBE system, the number of possible users (identities) does not have to be fixed at the beginning, thus we cannot really talk about full collusion resistance. If the number n of possible users was fixed, as in [10] for example, our construction would be fully collusion resistant.

2.3 Bilinear Maps

We briefly review the necessary facts about bilinear maps. Let $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ be three cyclic groups of prime order p. A bilinear map $e(\cdot,\cdot)$ is a map $\mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ such that for any generators $g_1 \in \mathbb{G}_1$, $g_2 \in \mathbb{G}_2$ and $a, b \in \mathbb{Z}_p$,

- $e\left({g_1}^a, {g_2}^b\right) = e\left(g_1, g_2\right)^{ab}$ (Bilinearity)
- $e\left(g_1, g_2\right) \neq 1$ (Non-degeneracy).

A bilinear map group system $\mathcal{B}$ is a tuple $\mathcal{B} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e(\cdot,\cdot))$, composed of objects as described above. $\mathcal{B}$ may also include group generators in its description. We impose all group operations as well as the bilinear map $e(\cdot,\cdot)$ to be efficiently computable, i.e. in time $\mathsf{poly}(|p|)$.

As seen later, we make use of an arbitrary bilinear map group system in our constructions. In particular, we do not need $\mathbb{G}_1$ and $\mathbb{G}_2$ to be distinct or equal. Neither do we require the existence of an efficient isomorphism going either way between $\mathbb{G}_1$ and $\mathbb{G}_2$, as it is the case for some pairing-based systems.

2.4 The General Diffie-Hellman Exponent Assumption

As in [15], we make use of the generalization of the Diffie-Hellman exponent assumption due to Boneh, Boyen and Goh [8]. They introduced a class of assumptions which includes a lot of assumptions that appeared with new pairing-based schemes. It includes for example DDH (in $\mathbb{G}_T$), BDH, $q-$BDHI, and $q-$BDHE assumptions.

We give an overview in the symmetric case. Let then $\mathcal{B} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e(\cdot,\cdot))$ be a bilinear map group system such that $\mathbb{G}_1 = \mathbb{G}_2 = \mathbb{G}$. Let $g_0 \in \mathbb{G}$ be a generator of $\mathbb{G}$, and set $g = e(g_0, g_0) \in \mathbb{G}_T$. Let s, n be positive integers and $P, Q \in \mathbb{F}_p[X_1, \ldots, X_n]^s$ be two s-tuples of n-variate polynomials over $\mathbb{F}_p$. Thus, P and Q are just two lists containing s multivariate polynomials each. We write $P = (p_1, p_2, \ldots, p_s)$ and $Q = (q_1, q_2, \ldots, q_s)$ and impose that $p_1 = q_1 = 1$. For any function $h : \mathbb{F}_p \to \Omega$ and vector $(x_1, \ldots, x_n) \in \mathbb{F}_p^n$, $h(P(x_1, \ldots, x_n))$ stands for $(h(p_1(x_1, \ldots, x_n)), \ldots, h(p_s(x_1, \ldots, x_n))) \in \Omega^s$. We use a similar notation for the s-tuple Q. Let $f \in \mathbb{F}_p[X_1, \ldots, X_n]$. It is said that f depends on (P, Q), which we denote by $f \in \langle P, Q \rangle$, when there exists a linear decomposition

$$f = \sum_{1 \leq i,j \leq s} a_{i,j} \cdot p_i \cdot p_j + \sum_{1 \leq i \leq s} b_i \cdot q_i \ , \qquad a_{i,j}, b_i \in \mathbb{Z}_p \ .$$

Let P, Q be as above and $f \in \mathbb{F}_p[X_1, \ldots, X_n]$. The (P, Q, f)-General Diffie-Hellman Exponent problems are defined as follows.

Definition 5 ((P, Q, f)-GDHE). *Given the tuple*

$$H(x_1, \ldots, x_n) = \left({g_0}^{P(x_1,\ldots,x_n)}, g^{Q(x_1,\ldots,x_n)}\right) \in \mathbb{G}^s \times \mathbb{G}_T^s \ ,$$

compute $g^{f(x_1,\ldots,x_n)}$.

Definition 6 (**(P, Q, f)-GDDHE**). *Given $H(x_1, \ldots, x_n) \in \mathbb{G}^s \times \mathbb{G}_T^s$ as above and $T \in \mathbb{G}_T$, decide whether $T = g^{f(x_1,\ldots,x_n)}$.*

We refer to [8] for a proof that (P, Q, f)-GDHE and (P, Q, f)-GDDHE have generic security when $f \notin \langle P, Q \rangle$. We will prove our constructions are secure based on the assumption that (P, Q, f)-GDDHE is intractable for any $f \notin \langle P, Q \rangle$ and polynomial parameters $s, n = \mathsf{poly}(\lambda)$. We just have to determine P, Q and f, such that we can perform our simulation, and then proving the condition on the polynomials will prove the intractability of our problem (because as seen before, the (P, Q, f)-GDDHE problem is hard for any choice of P, Q and f which satisfy the aforementioned condition).

3 Our Construction

3.1 Description

In this section, we present our new IBBE, with constant size ciphertexts and private keys.

$\mathsf{Setup}(\lambda, m)$. Given the security parameter λ and an integer m, a bilinear map group system $\mathcal{B} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e(\cdot,\cdot))$ is constructed such that $|p| = \lambda$. Also, two generators $g \in \mathbb{G}_1$ and $h \in \mathbb{G}_2$ are randomly selected as well as a secret value $\gamma \in \mathbb{Z}_p^\star$. Choose a cryptographic hash function $\mathcal{H} : \{0,1\}^\star \to \mathbb{Z}_p^\star$. The security analysis will view $\mathcal{H}$ as a random oracle. $\mathcal{B}$ and $\mathcal{H}$ constitute system public parameters. The master secret key is defined as $\mathsf{MSK} = (g, \gamma)$. The public key is $\mathsf{PK} = \left(w, v, h, h^\gamma, \ldots, h^{\gamma^m}\right)$ where $w = g^\gamma$, and $v = e(g, h)$.

$\mathsf{Extract}(\mathsf{MSK}, \mathsf{ID})$. Given $\mathsf{MSK} = (g, \gamma)$ and the identity ID, it outputs

$$\mathsf{sk}_{\mathsf{ID}} = g^{\frac{1}{\gamma + \mathcal{H}(\mathsf{ID})}}$$

$\mathsf{Encrypt}(\mathcal{S}, \mathsf{PK})$. Assume for notational simplicity that $\mathcal{S} = \{\mathsf{ID}_j\}_{j=1}^s$, with $s \leq m$. Given $\mathsf{PK} = \left(w, v, h, h^\gamma, \ldots, h^{\gamma^m}\right)$, the broadcaster randomly picks $k \leftarrow \mathbb{Z}_p^\star$ and computes $\mathsf{Hdr} = (C_1, C_2)$ and K where

$$C_1 = w^{-k}\ , \qquad C_2 = h^{k \cdot \prod_{i=1}^s (\gamma + \mathcal{H}(\mathsf{ID}_i))}\ , \qquad K = v^k\ .$$

$\mathsf{Encrypt}$ outputs (Hdr, K). (Then K is used to encrypt the message)

$\mathsf{Decrypt}(\mathcal{S}, \mathsf{ID}_i, \mathsf{sk}_{\mathsf{ID}_i}, \mathsf{Hdr}, \mathsf{PK})$. In order to retrieve the message encryption key K encapsulated in the header $\mathsf{Hdr} = (C_1, C_2)$, user with identity ID_i and the corresponding private key $\mathsf{sk}_{\mathsf{ID}_i} = g^{\frac{1}{\gamma + \mathcal{H}(\mathsf{ID}_i)}}$ (with $\mathsf{ID}_i \in \mathcal{S}$) computes

$$K = \left(e\left(C_1, h^{p_{i,\mathcal{S}}(\gamma)}\right) \cdot e\left(\mathsf{sk}_{\mathsf{ID}_i}, C_2\right)\right)^{\frac{1}{\prod_{j=1, j \neq i}^s \mathcal{H}(\mathsf{ID}_j)}}$$

with

$$p_{i,\mathcal{S}}(\gamma) = \frac{1}{\gamma} \cdot \left(\prod_{j=1, j\neq i}^{s} (\gamma + \mathcal{H}(\mathsf{ID}_j)) - \prod_{j=1, j\neq i}^{s} \mathcal{H}(\mathsf{ID}_j)\right)$$

Correctness: Assuming C is well-formed for $\mathcal{S}$:

$$\begin{aligned}
K' &:= e\left(C_1, h^{p_{i,\mathcal{S}}(\gamma)}\right) \cdot e\left(\mathsf{sk}_{\mathsf{ID}i}, C_2\right) \\
&= e\left(g^{-k.\gamma}, h^{p_{i,\mathcal{S}}(\gamma)}\right) \cdot e\left(g^{\frac{1}{\gamma+\mathcal{H}(\mathsf{ID}_i)}}, h^{k\cdot\prod_{j=1}^{s}(\gamma+\mathcal{H}(\mathsf{ID}_j))}\right) \\
&= e\left(g,h\right)^{-k\cdot\left(\prod_{j=1,j\neq i}^{s}(\gamma+\mathcal{H}(\mathsf{ID}_j))-\prod_{j=1,j\neq i}^{s}\mathcal{H}(\mathsf{ID}_j)\right)} \cdot e\left(g,h\right)^{k\cdot\prod_{j=1,j\neq i}^{s}(\gamma+\mathcal{H}(\mathsf{ID}_j))} \\
&= e\left(g,h\right)^{k\prod_{j=1,j\neq i}^{s}\mathcal{H}(\mathsf{ID}_j)} \\
&= K^{\prod_{j=1,j\neq i}^{s}\mathcal{H}(\mathsf{ID}_j)}
\end{aligned}$$

Thus ${K'}^{\frac{1}{\prod_{j=1,j\neq i}^{s}\mathcal{H}(\mathsf{ID}_j)}} = K$.

Efficiency. Our construction achieves $O(1)$-size ciphertexts, $O(m)$-size public keys and constant size private keys. Note that public key is linear in the maximal size of $\mathcal{S}$, and not in the number of decryption keys that can be distributed. If we would like to fix the total number n of users, and set $m = n$, then we would reduce the public key size by a factor of two from BGW. Note also that as we said before, the broadcaster has to send the set $\mathcal{S}$ of identities that are included in the ciphertext. This set is needed to decrypt, as in previous schemes, thus it is counted in the full header, but not in the header.

3.2 Security Analysis

We prove the IND-sID-CPA security of our system by using the GDDHE framework of [8]. We start by defining the following intermediate decisional problem.

Definition 7 ((f, g, F)-GDDHE). *Let $\mathcal{B} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e(\cdot,\cdot))$ be a bilinear map group system and let f and g be two coprime polynomials with pairwise distinct roots, of respective orders t and n. Let g_0 be a generator of $\mathbb{G}_1$ and h_0 a generator of $\mathbb{G}_2$. Solving the (f, g, F)-GDDHE problem consists, given*

$$\begin{array}{lll}
g_0\ , {g_0}^{\gamma}\ , \ldots , {g_0}^{\gamma^{t-1}}\ , & {g_0}^{\gamma\cdot f(\gamma)}\ , & {g_0}^{k\cdot\gamma\cdot f(\gamma)}\ , \\
h_0\ , {h_0}^{\gamma}\ , \ldots , {h_0}^{\gamma^{2n}}\ , & & {h_0}^{k\cdot g(\gamma)}\ ,
\end{array}$$

and $T \in \mathbb{G}_T$, in deciding whether T is equal to $e(g_0, h_0)^{k\cdot f(\gamma)}$ or to some random element of $\mathbb{G}_T$.

We denote by $\mathsf{Adv}^{\mathsf{gddhe}}(f, g, F, \mathcal{A})$ the advantage of an algorithm $\mathcal{A}$ in distinguishing the two distributions and set $\mathsf{Adv}^{\mathsf{gddhe}}(f, g, F) = \max_{\mathcal{A}} \mathsf{Adv}^{\mathsf{gddhe}}(f, g, F, \mathcal{A})$ over $\mathsf{poly}(|p|)$-time $\mathcal{A}$'s.

The following statement is a corollary of Theorem 2 which can be found in Appendix A. This corollary concerns the case where the polynomials are of the form described above (see the reformulation of the problem in Appendix A).

Corollary 1 (Generic security of (f,g,F)-GDDHE). *For any probabilistic algorithm $\mathcal{A}$ that totalizes of at most q queries to the oracles performing the group operations in $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ and the bilinear map $e(\cdot,\cdot)$,*

$$\mathsf{Adv}^{\mathsf{gddhe}}(f,g,F,\mathcal{A}) \leq \frac{(q+2(n+t+4)+2)^2 \cdot d}{2p}$$

with $d = 2 \cdot \max(n, t+1)$.

IND-sID-CPA Security. Let $\mathcal{IBBE}$ denote our construction as per Section 3. We state:

Theorem 1. *For any n, t, we have $\mathsf{Adv}^{\mathsf{ind}}_{\mathcal{IBBE}}(t,n) \leq 2 \cdot \mathsf{Adv}^{\mathsf{gddhe}}(f,g,F)$.*

The rest of this section is dedicated to proving Theorem 1. To establish the semantic security of $\mathcal{IBBE}$ against static adversaries, we assume to be given an adversary $\mathcal{A}$ breaking it under a (t,n)-collusion and we build a reduction algorithm $\mathcal{R}$ that distinguishes the two distributions of the (f,g,F)-GDDHE problem.

Both the adversary and the challenger are given as input n, the maximal size of a set of included users $\mathcal{S}$, and t the total number of extraction queries and random oracle queries that can be issued by the adversary.

Algorithm $\mathcal{R}$ is given as input a group system $\mathcal{B} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e(\cdot,\cdot))$, and a (f,g,F)-GDDHE instance in $\mathcal{B}$ (as described in Definition 7). We thus have f and g two coprime polynomials with pairwise distinct roots, of respective orders t and n, and $\mathcal{R}$ is given

$$\begin{array}{lll} g_0\,, {g_0}^{\gamma}\,, \ldots, {g_0}^{\gamma^{t-1}}\,, & {g_0}^{\gamma \cdot f(\gamma)}\,, & {g_0}^{k \cdot \gamma \cdot f(\gamma)}\,, \\ h_0\,, {h_0}^{\gamma}\,, \ldots, {h_0}^{\gamma^{2n}}\,, & & {h_0}^{k \cdot g(\gamma)}\,, \end{array}$$

as well as $T \in \mathbb{G}_T$ which is either equal to $e(g_0, h_0)^{k \cdot f(\gamma)}$ or to some random element of $\mathbb{G}_T$.

For simplicity, we state that f and g are unitary polynomials, but this is not a mandatory requirement.

Notations

- $f(X) = \prod_{i=1}^{t}(X + x_i)$, $g(X) = \prod_{i=t+1}^{t+n}(X + x_i)$
- $f_i(x) = \frac{f(x)}{x+x_i}$ for $i \in [1,t]$, which is a polynomial of degree $t-1$
- $g_i(x) = \frac{g(x)}{x+x_i}$ for $i \in [t+1, t+n]$, which is a polynomial of degree $n-1$

Init: The adversary $\mathcal{A}$ outputs a set $\mathcal{S}^* = \{\mathsf{ID}_1^*, \ldots, \mathsf{ID}_{s^*}^*\}$ of identities that he wants to attack (with $s^* \leq n$).

Setup: To generate the system parameters, $\mathcal{R}$ formally sets $g = {g_0}^{f(\gamma)}$ (i.e. without computing it) and sets

$$h = {h_0}^{\prod_{i=t+s^*+1}^{t+n}(\gamma+x_i)} , \qquad w = {g_0}^{\gamma\cdot f(\gamma)} = g^{\gamma} ,$$

$$v = e\left(g_0, h_0\right)^{f(\gamma)\cdot\prod_{i=t+s^*+1}^{t+n}(\gamma+x_i)} = e\left(g, h\right) .$$

$\mathcal{R}$ then defines the public key as $\mathsf{PK} = \left(w, v, h, h^{\gamma}, \ldots, h^{\gamma^n}\right)$. Note that $\mathcal{R}$ can by no means compute the value of g. $\mathcal{R}$ runs $\mathcal{A}$ on the system parameters $(\mathcal{B}, \mathcal{H})$ and PK, with $\mathcal{H}$ a random oracle controlled by $\mathcal{R}$ described below.

Hash Queries: At any time the adversary $\mathcal{A}$ can query the random oracle on any identity ID_i (at most $t - q_E$ times, with q_E the number of extraction queries). To respond to these queries, $\mathcal{R}$ maintains a list $\mathcal{L}_{\mathcal{H}}$ of tuples $(\mathsf{ID}_i, x_i, \mathsf{sk}_{\mathsf{ID}_i})$ that contains at the beginning:

$$\left\{(*, x_i, *)\right\}_{i=1}^{t} , \quad \left\{(\mathsf{ID}_i, x_i, *)\right\}_{i=t+1}^{t+s^*}$$

(we choose to note "$*$" an empty entry in $\mathcal{L}_{\mathcal{H}}$). When the adversary issues a hash query on identity ID_i,

1. If ID_i already appears in the list $\mathcal{L}_{\mathcal{H}}$, $\mathcal{R}$ responds with the corresponding x_i.
2. Otherwise, $\mathcal{R}$ sets $\mathcal{H}(\mathsf{ID}_i) = x_i$, and completes the list with $(\mathsf{ID}_i, x_i, *)$.

Query phase 1: The adversary $\mathcal{A}$ adaptively issues queries $q_1, \ldots, q_m$, where q_i is an Extraction query (ID_i): The challenger runs $\mathsf{Extract}$ on $\mathsf{ID}_i \notin \mathcal{S}^*$ and forwards the resulting private key to the adversary. To generate the keys,

- if $\mathcal{A}$ has already issued an extraction query on ID_i, $\mathcal{R}$ responds with the corresponding $\mathsf{sk}_{\mathsf{ID}_i}$ in the list $\mathcal{L}_{\mathcal{H}}$.
- else, if $\mathcal{A}$ has already issued a hash query on ID_i, then $\mathcal{R}$ uses the corresponding x_i to compute

$$\mathsf{sk}_{\mathsf{ID}_i} = {g_0}^{f_i(\gamma)} = g^{\frac{1}{\gamma+\mathcal{H}(\mathsf{ID}_i)}}$$

One can verify that $\mathsf{sk}_{\mathsf{ID}_i}$ is a valid private key. $\mathcal{R}$ then completes the list $\mathcal{L}_{\mathcal{H}}$ with $\mathsf{sk}_{\mathsf{ID}_i}$ for ID_i.

1. Otherwise, $\mathcal{R}$ sets $\mathcal{H}(\mathsf{ID}_i) = x_i$, computes the corresponding $\mathsf{sk}_{\mathsf{ID}_i}$ exactly as above, and completes the list $\mathcal{L}_{\mathcal{H}}$ for ID_i.

Challenge: When $\mathcal{A}$ decides that phase 1 is over, algorithm $\mathcal{R}$ computes $\mathsf{Encrypt}$ to obtain $(\mathsf{Hdr}^*, K) = \mathsf{Encrypt}(\mathcal{S}^*, \mathsf{PK})$

$$C_1 = {g_0}^{-k\cdot\gamma\cdot f(\gamma)} , \quad C_2 = {h_0}^{k\cdot g(\gamma)} , \quad K = T^{\prod_{i=t+s^*+1}^{t+n} x_i} \cdot e\left({g_0}^{k\cdot\gamma\cdot f(\gamma)}, {h_0}^{q(\gamma)}\right)$$

with $q(\gamma) = \frac{1}{\gamma} \cdot \left(\prod_{i=t+s^*+1}^{t+n}(\gamma + x_i) - \prod_{i=t+s^*+1}^{t+n} x_i\right)$.

One can verify that:

$$C_1 = w^{-k} , \quad C_2 = {h_0}^{k\cdot\prod_{i=t+s^*+1}^{t+n}(\gamma+x_i)\cdot\prod_{i=t+1}^{t+s^*}(\gamma+x_i)} = h^{k\cdot\prod_{i=t+1}^{t+s^*}(\gamma+\mathcal{H}(\mathsf{ID}_i^*))} .$$

Note that if $T = e\left(g_0, h_0\right)^{k\cdot f(\gamma)}$, then $K = v^k$.

The challenger then randomly selects $b \leftarrow \{0,1\}$, sets $K_b = K$, and sets K_{1-b} to a random value in $\mathcal{K}$. The challenger returns $(\mathsf{Hdr}^*, K_0, K_1)$ to $\mathcal{A}$.

Query phase 2: The adversary continues to issue queries $q_{m+1}, \ldots, q_E$ where q_i is an extraction query (ID_i) with the constraint that $\mathsf{ID}_i \notin \mathcal{S}^*$ (identical to phase 1).

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

One has

$$\begin{aligned}\mathsf{Adv}^{\mathsf{gddhe}}(f,g,F,\mathcal{R}) &= \Pr[b' = b|\,\mathsf{real}] - \Pr[b' = b|\,\mathsf{rand}] \\ &= \frac{1}{2} \times (\Pr[b' = 1|b = 1 \wedge \mathsf{real}] - \Pr[b' = 1|b = 0 \wedge \mathsf{real}]) \\ &- \frac{1}{2} \times (\Pr[b' = 1|b = 1 \wedge \mathsf{rand}] + \Pr[b' = 1|b = 0 \wedge \mathsf{rand}])\,.\end{aligned}$$

Now in the random case, the distribution of b is independent from the adversary's view wherefrom

$$\Pr[b' = 1|b = 1 \wedge \mathsf{rand}] = \Pr[b' = 1|b = 0 \wedge \mathsf{rand}]\ .$$

In the real case however, the distributions of all variables defined by $\mathcal{R}$ perfectly comply with the semantic security game since all simulations are perfect. Therefore

$$\mathsf{Adv}^{\mathsf{ind}}_{\mathcal{IBBE}}(t,n,\mathcal{A}) = \Pr[b' = 1|b = 1 \wedge \mathsf{real}] - \Pr[b' = 1|b = 0 \wedge \mathsf{real}]\ .$$

Putting it altogether, we get that $\mathsf{Adv}^{\mathsf{gddhe}}(f,g,F,\mathcal{R}) = \frac{1}{2} \cdot \mathsf{Adv}^{\mathsf{ind}}_{\mathcal{IBBE}}(t,n,\mathcal{A})$.

Remark. Note that if the attacker makes less key derivation queries than random oracle queries, we generate keys that we never give out, but this is not a problem.

About chosen-ciphertext attacks. The Cannetti, Halevi, and Katz [12] result applies here. Just making one of the identities that we broadcast to derive from a verification key of a strong signature scheme. Then it can be used to sign the ciphertext.

Removing the Random Oracle Model. One way to remove the random oracle model could be to randomize the private key extraction as follows: For an identity ID_i, $\mathsf{sk}_{\mathsf{ID}_i} = g^{\frac{1}{\gamma+\mathsf{ID}_i}}$ could be replaced by $A_i = g^{\frac{1}{\gamma+\mathsf{ID}_i+r_i\cdot\alpha}}$, with α an element of MSK and r_i chosen by the $\mathcal{PKG}$. Note that this randomization has already been employed in [6].

Note also that we could easily obtain IND-na-sID-CPA *without* random oracles by using an assumption which is not fully non-interactive. Indeed, during the setup, if the algorithm is given a (f,g,F)-GDDHE instance, with g that corresponds to the target set and f to the corrupted set (chosen by the attacker at initialization), then the rest of the proof can be done without any oracle.

4 Conclusion

We introduced the first identity-based broadcast encryption (IBBE) scheme with constant size ciphertexts and private keys. One interesting open problem would be to construct an IBBE system with constant size ciphertexts and private keys that is secure under a more standard assumption, or which achieves a stronger security notion, equivalent to full security in IBE schemes.

Acknowledgements

The author would like to thank David Pointcheval, Pascal Paillier and Brent Waters for helpful discussions, and anonymous referees for helpful comments.

References

1. Abdalla, M., Kiltz, E., Neven, G.: Generalized key delegation for hierarchical identity-based encryption. In: ESORICS 2007. LNCS, vol. 4734, pp. 139–154. Springer, Berlin, Germany (2005)
2. Baek, J., Safavi-Naini, R., Susilo, W.: Efficient multi-receiver identity-based encryption and its application to broadcast encryption. In: Vaudenay, S. (ed.) PKC 2005. LNCS, vol. 3386, pp. 380–397. Springer, Heidelberg (2005)
3. Barbosa, M., Farshim, P.: Efficient identity-based key encapsulation to multiple parties. In: Smart, N.P. (ed.) Cryptography and Coding. LNCS, vol. 3796, pp. 428–441. Springer, Heidelberg (2005)
4. Bellare, M., Boldyreva, A., Micali, S.: Public-key encryption in a multi-user setting: Security proofs and improvements. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 259–274. Springer, Berlin, Germany (2000)
5. Bellare, M., Boldyreva, A., Staddon, J.: Randomness re-use in multi-recipient encryption schemeas. In: Desmedt, Y.G. (ed.) PKC 2003. LNCS, vol. 2567, pp. 85–99. Springer, Heidelberg (2002)
6. Boneh, D., Boyen, X.: Efficient selective-ID secure identity based encryption without random oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 223–238. Springer, Berlin, Germany (2004)
7. Boneh, D., Boyen, X.: Secure identity based encryption without random oracles. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, Springer, Berlin, Germany (2004)
8. Boneh, D., Boyen, X., Goh, E.-J.: Hierarchical identity based encryption with constant size ciphertext. In: Cramer, R.J.F. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 440–456. Springer, Heidelberg (2005), available at `http://eprint.iacr.org/2005/015`
9. Boneh, D., Franklin, M.: Identity-based encryption from the Weil pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Berlin, Germany (2001)

10. 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, Berlin, Germany (2005)
11. Canetti, R., Halevi, S., Katz, J.: A forward-secure public-key encryption scheme. In: Biham, E. (ed.) Advances in Cryptology – EUROCRPYT 2003. LNCS, vol. 2656, pp. 255–271. Springer, Berlin, Germany (2003)
12. Canetti, R., Halevi, S., Katz, J.: Chosen-ciphertext security from identity-based encryption. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 207–222. Springer, Berlin, Germany (2004)
13. Chatterjee, S., Sarkar, P.: Multi-receiver identity-based key encapsulation with shortened ciphertext. In: Barua, R., Lange, T. (eds.) INDOCRYPT 2006. LNCS, vol. 4329, pp. 394–408. Springer, Heidelberg (2006)
14. Cocks, C.: An identity based encryption scheme based on quadratic residues. In: Honary, B. (ed.) Cryptography and Coding. LNCS, vol. 2260, pp. 360–363. Springer, Berlin, Germany (2001)
15. Delerablée, C., Paillier, P., Pointcheval, D.: Fully collusion secure dynamic broadcast encryption with constant-size ciphertexts or decryption keys. In: Takagi, T., et al. (eds.) PAIRING 2007. LNCS, vol. 4575, pp. 39–59. Springer, Berlin, Germany (2007)
16. Fiat, A., Naor, M.: Broadcast encryption. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 480–491. Springer, Heidelberg (1994)
17. Gentry, C.: Practical identity-based encryption without random oracles. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 445–464. Springer, Berlin, Germany (2006)
18. Gentry, C., Silverberg, A.: Hierarchical ID-based cryptography. In: Zheng, Y. (ed.) ASIACRYPT 2002. LNCS, vol. 2501, pp. 548–566. Springer, Berlin, Germany (2002)
19. 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)
20. Halevy, D., Shamir, A.: The LSD broadcast encryption scheme. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 47–60. Springer, Heidelberg (2002)
21. Horwitz, J., Lynn, B.: Toward hierarchical identity-based encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 466–481. Springer, Heidelberg (2002)
22. Kurosawa, K.: Multi-recipient public-key encryption with shortened ciphertext. In: Naccache, D., Paillier, P. (eds.) PKC 2002. LNCS, vol. 2274, pp. 48–63. Springer, Heidelberg (2002)
23. 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, Berlin, Germany (2001)
24. Shamir, A.: Identity-based cryptosystems and signature schemes. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985)
25. Smart, N.P.: Efficient key encapsulation to multiple parties. In: Blundo, C., Cimato, S. (eds.) SCN 2004. LNCS, vol. 3352, pp. 208–219. Springer, Heidelberg (2005)
26. Brent, R.: Efficient identity-based encryption without random oracles. In: Cramer, R.J.F. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Berlin, Germany (2005)

A Intractability of (f, g, F)-GDDHE

In this section, we prove the intractability of distinguishing the two distributions involved in the (f, g, F)-GDDHE problem (cf. Corollary 1, section 3.2). We first review some results on the General Diffie-Hellman Exponent Problem, from [8]. In order to be the most general, we assume the easiest case for the adversary: when $\mathbb{G}_1 = \mathbb{G}_2$, or at least that an isomorphism that can be easily computed in either one or both ways is available.

Theorem 2 ([8]). *Let $P, Q \in \mathbb{F}_p[X_1, \ldots, X_m]$ be two s-tuples of m-variate polynomials over $\mathbb{F}_p$ and let $F \in \mathbb{F}_p[X_1, \ldots, X_m]$. Let d_P (resp. d_Q, d_F) denote the maximal degree of elements of P (resp. of Q, F) and pose $d = \max(2d_P, d_Q, d_F)$. If $F \notin \langle P, Q\rangle$ then for any generic-model adversary $\mathcal{A}$ totalizing at most q queries to the oracles (group operations in $\mathbb{G}, \mathbb{G}_T$ and evaluations of e) which is given $H(x_1, \ldots, x_m)$ as input and tries to distinguish $g^{F(x_1,\ldots,x_m)}$ from a random value in $\mathbb{G}_T$, one has*

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

Proof (of Corollary 1). In order to conclude with Corollary 1, we need to prove that the (f, g, F)-GDDHE problem lies in the scope of Theorem 2. As already said, we consider the weakest case $\mathbb{G}_1 = \mathbb{G}_2 = \mathbb{G}$ and thus pose $h_0 = {g_0}^{\beta}$. Our problem can be reformulated as (P, Q, F)-GDHE where

$$\begin{aligned} P &= \begin{pmatrix} 1, \gamma, \gamma^2, \ldots, \gamma^{t-1}, & \gamma \cdot f(\gamma), k \cdot \gamma \cdot f(\gamma) \\ \beta, \beta \cdot \gamma, \beta \cdot \gamma^2, \ldots, \beta \cdot \gamma^{2n}, & k \cdot \beta \cdot g(\gamma) \end{pmatrix} \\ Q &= 1 \\ F &= k \cdot \beta \cdot f(\gamma), \end{aligned}$$

and thus $m = 3$ and $s = t + n + 4$. We have to show that F is independent of (P, Q), i.e. that no coefficients $\{a_{i,j}\}_{i,j=1}^{s}$ and b_1 exist such that $F = \sum_{i,j=1}^{s} a_{i,j} p_i p_j + \sum_{k=1}^{2} b_1 q_1$ where the polynomials p_i and q_1 are the one listed in P and Q above. By making all possible products of two polynomials from P which are multiples of $k \cdot \beta$, we want to prove that no linear combination among the polynomials from the list R below leads to F:

$$R = \begin{pmatrix} k \cdot \beta \cdot \gamma \cdot f(\gamma),\ k \cdot \beta \cdot \gamma^2 \cdot f(\gamma), \ldots,\ k \cdot \beta \cdot \gamma^{n+1} \cdot f(\gamma), \\ k \cdot \beta \cdot g(\gamma),\ k \cdot \beta \cdot \gamma \cdot g(\gamma), \ldots,\ k \cdot \beta \cdot \gamma^{t-1} \cdot g(\gamma) \\ k \cdot \beta \cdot \gamma \cdot f(\gamma) g(\gamma) \end{pmatrix} .$$

Note that the last polynomial can be written as $k \cdot \beta \cdot \gamma \cdot f(\gamma) g(\gamma) = \sum_{i=0}^{i=n} \nu_i \cdot k \cdot \beta \cdot \gamma^{i+1} \cdot f(\gamma)$, and thus as a linear combination of the polynomials from the first line. We therefore simplify the task to refuting a linear combination of elements of the list R' below which leads to $f(\gamma)$:

$$R' = \begin{pmatrix} \gamma \cdot f(\gamma),\ \gamma^2 \cdot f(\gamma), \ldots,\ \gamma^{n+1} \cdot f(\gamma), \\ g(\gamma),\ \gamma \cdot g(\gamma), \ldots,\ \gamma^{t-1} \cdot g(\gamma) \end{pmatrix} .$$

Any such linear combination can be written as

$$f(\gamma) = A(\gamma) \cdot f(\gamma) + B(\gamma) \cdot g(\gamma)$$

where A and B are polynomials such that $A(0) = 0$, $\deg A \leq n+1$ and $\deg B \leq t-1$. Since f and g are coprime by assumption, we must have $f \mid B$. Since $\deg f = t$ and $\deg B \leq t-1$ this implies $B = 0$. Hence $A = 1$ which contradicts $A(0) = 0$. □