This Journal Version appears
in International Journal of Information Security, Volume 12, Number 4, pages 251–265. 2013.

Adaptive CCA Broadcast Encryption with Constant-Size Secret Keys and Ciphertexts

Duong-Hieu Phan[1], David Pointcheval[2], Siamak F. Shahandashti[1], and Mario Strefler[2]

[1] Université de Paris 8, France
[2] École Normale Supérieure–CNRS–INRIA, Paris, France

Abstract. We consider designing public-key broadcast encryption schemes with constant-size secret keys and ciphertexts, achieving chosen-ciphertext security. We first argue that known CPA-to-CCA transforms currently do not yield such schemes. We then propose a scheme, modifying a previous selective CPA secure proposal by Boneh, Gentry, and Waters. Our scheme has constant-size secret keys and ciphertexts and we prove that it is selective chosen-ciphertext secure based on standard assumptions. Our scheme has ciphertexts that are shorter than those of the previous CCA secure proposals. Then we propose a second scheme that provides the functionality of both broadcast encryption and revocation schemes simultaneously using the same set of parameters. Finally we show that it is possible to prove our first scheme adaptive chosen-ciphertext secure under reasonable extensions of the bilinear Diffie-Hellman exponent and the knowledge of exponent assumptions. We prove both of these extended assumptions in the generic group model. Hence, our scheme becomes the first to achieve constant-size secret keys and ciphertexts (both asymptotically optimal) and adaptive chosen-ciphertext security at the same time.

1 Introduction

A *broadcast encryption* is a cryptographic scheme that enables encryption of broadcast content such that only a set of target users, selected at the time of encryption, can decrypt the content. Apparent applications include group communication, pay TV, content protection, file system access control, and geolocation.

A crucial aspect of any cryptographic scheme, which arguably decides its fate of being used in practice, is its efficiency. Since one of the most prominent applications of broadcast encryption is real-time broadcasting, ciphertext size is at the heart of efficiency measures for such schemes, and constructions with constant-size ciphertexts are desirable. Indeed, if one allows the ciphertext size to grow linearly with the number of target users, construction of secure broadcast encryption becomes trivial. Other important measures of efficiency for broadcast encryption include the secret and public key sizes and the encryption and decryption times.

A broadcast encryption scheme can be *static* or *dynamic*, depending on if the system users need to be fixed once and for all at the setup stage or if it supports new users joining the system at an arbitrary time, incurring only incremental parameter changes. Evidently, dynamic schemes are more flexible and hence more desirable in practical applications.

An important security paradigm for broadcast encryption schemes is that of *adaptive* security. This paradigm captures the fact that an adversary might choose to compromise keys in the system adaptively, based on its acquired knowledge of the system parameters and previously compromised keys and ciphertexts. Such a definition is widely accepted as the proper notion of security for broadcast encryption schemes and there are schemes proposed in the literature that provably achieve security against adaptive adversaries.

On the other hand, security against chosen ciphertext attacks (CCA) is a fundamental notion of security for any encryption scheme, broadcast encryption included. Although there have been a number of proposed broadcast encryption schemes that are secure against chosen plaintext attacks (CPA), the CPA-to-CCA transformations in the literature do not seem to yield CCA secure broadcast encryption schemes with constant-size ciphertexts.

Adaptive and CCA security, and constant-size ciphertexts, have all three been separately achieved for broadcast encryption. However, there has not been any proposal that achieves all three simultaneously. In this paper, we propose a broadcast encryption with constant-size ciphertexts and prove it

adaptive CCA secure under assumptions that are reasonable generalizations of previous assumptions in the literature.

The literature on broadcast encryption mainly considers two categories of such schemes and each work usually provides solutions that are efficient only for one of the two cases, depending on whether the content is broadcast to a very small or a very large proportion of registered users. The party who encrypts the content, hence either determines their intended set of target users or that of revoked users, respectively, as an input to the encryption algorithm. Consequently, the latter category of schemes are sometimes called *revocation* schemes.

Consider the pay-TV application in which the content of the broadcast consists of several TV channels. Normally, there are a number of basic channels that are usually bundled together and provided to most of the customers in different packages, and also there are a number of more specialized channels (e.g., pay-per-view) that are of the interest of a small proportion of customers. Hence we face a scenario in which both of the above categories of schemes are simultaneously needed to broadcast the content. Nevertheless, there has been no proposal in the literature that provides both functionalities efficiently, and hence the existing efficient solution to the above scenario is to set up two parallel schemes, each covering part of the broadcast content. In this paper, we propose a scheme that can handle both cases efficiently, providing a solution to the above scenario that does not require maintaining two parallel sets of system parameters.

1.1 Related Work

Broadcast encryption was first formalized by Fiat and Naor [FN94]. Their scheme is a private key scheme and proved secure against an upper bounded number of colluders. Fully collusion secure (private-key) broadcast encryption was first proposed in [NNL01], which introduced the subset cover framework that became the basis for many subsequent proposals, including [DF03] which proposed the first public key broadcast encryption.

Boneh, Gentry, and Waters [BGW05] were first to propose a fully collusion-resistant public key broadcast encryption in which the ciphertext size is constant. In all the previous schemes, the size of the ciphertext is linear in the size of the target set. In this paper we limit our attention to such schemes. They proposed two schemes, respectively CPA and CCA secure, both in the selective model of security. Dynamic broadcast encryption was proposed in [DPP07] where they designed CPA secure schemes that were only partially adaptive secure. Strictly speaking, their scheme is a *revocation* scheme, in which the set of revoked users is selected at the time of encryption, and in turn, any user outside of the revoked set is able to decrypt. [Del07] proposed identity-based broadcast encryption and gave a selective CPA secure scheme.

Adaptive security was first proposed by [GW09] where they gave several schemes achieving adaptive CPA security, including two broadcast encryption schemes and two identity-based broadcast encryption (IBBE) schemes, one of each achieving constant-size ciphertexts in the random oracle model. The schemes proposed in [Wat09] and [LSW10], respectively a broadcast encryption scheme and a revocation scheme, are the only schemes secure under static assumptions (as opposed to the so called q-based ones). The latter work also proposes an identity-based revocation scheme which is proved selective CPA secure. Recently, the first adaptive CCA secure schemes were proposed by [PPS11a], although their schemes do not have constant-size ciphertexts.

1.2 Our Contributions

In this paper, we propose an efficient dynamic broadcast encryption scheme (called OurBE) and prove that it is selective CCA secure assuming the widely-used bilinear Diffie-Hellman exponent (BDHE) assumption and a universal one-way hash function (UOWHF). The scheme has constant-size ciphertexts (only two group elements), constant-size secret keys (only one group element), and a public key which grows linearly with the number of users in the system. We construct our scheme by modifying a selective CPA secure scheme (dubbed BGW_1 from now on) by Boneh, Gentry, and Waters [BGW05].

Our modification is minimal in the sense that our scheme has exactly the same ciphertext and secret key sizes as that of BGW_1, and is proved secure under the same assumption, plus the comparatively weak UOWHF assumption. The minor difference is that our scheme has one extra element in the linearly-growing public key. The only other CCA secure scheme with constant-size ciphertexts is a modified version of BGW_1 by the same authors (dubbed BGW_2 from now on), which has ciphertexts that are double the size of our scheme (i.e., four group elements vs. our two). BGW_2 is proved selective CCA secure under BDHE, plus the assumption that a signature scheme used in the construction is strongly unforgeable, which is an assumption of comparable strength as UOWHF.

We also propose an *inclusive-exclusive* broadcast encryption scheme which can act as both a broadcast encryption and a revocation scheme at the same time, as it allows the flexibility to specify either the target set or the revoked set at the time of encryption. The ciphertext and the secret key are still only two and one group elements, respectively, but we need to add one group element per user to the already linearly-growing public key which results in a public key which is 1.5 times that of BGW_1.

Next, we show that it is possible to prove OurBE adaptive CCA secure under generalized versions of existing assumptions. Particularly, we propose generalized versions of the BDHE and the knowledge-of-exponent (KEA) assumptions, and prove that both hold in the generic group model. We argue that both of these are intuitive and reasonable generalizations of accepted assumptions, and in turn, enable achieving the highest level of security with highly-efficient parameters. Namely, OurBE is provably adaptive CCA secure with constant-size ciphertexts and secret keys, and it is the first scheme to achieve such properties.

2 Preliminaries

In this section we review the notation we use, the BDHE and GBDHE assumptions, and the notions of security for dynamic broadcast encryption and universal one-way hash function.

Notation We use the following typefaces: Roman X for constants, italic X for variables, sans serif X for algorithms, and calligraphic $\mathcal{X}$ for oracles. Let $\mathbb{G}$ and $\mathbb{G}_\mathrm{T}$ be groups of order p, and $e : \mathbb{G} \times \mathbb{G} \mapsto \mathbb{G}_\mathrm{T}$ be a bilinear map. Let g be a generator of $\mathbb{G}$ and $g_\mathrm{T} = e(g, g)$.

2.1 Dynamic Broadcast Encapsulation

Broadcast encryption is conventionally formalized as *broadcast encapsulation* in which, instead of a ciphertext, a session key is produced, which is required to be indistinguishable from random. Such a scheme can provide public encryption functionality in combination with a symmetric encryption through the hybrid encryption (a.k.a. KEM-DEM) paradigm [CS03]. We hence use the terms encryption and encapsulation interchangeably.

Following [DPP07], we define a (public-key) *dynamic* broadcast encapsulation scheme as a tuple of four algorithms $\mathsf{BE} = (\mathsf{Setup}, \mathsf{Join}, \mathsf{Encaps}, \mathsf{Decaps})$ where:

- $\mathsf{Setup}(1^k)$ outputs (MSK, EK) containing the master secret key and the (initial) encryption key;
- $\mathsf{Join}(MSK, i)$ outputs the key pair (sk_i, pk_i) for user i, and appends pk_i to EK;
- $\mathsf{Encaps}(EK, S)$ for a set of users S outputs (H, K) containing a ciphertext (a.k.a. header) and a session key; and
- $\mathsf{Decaps}(EK, sk_i, S, H)$ outputs K if $i \in S$ and $\perp$ otherwise.

Adaptive CCA security for BE is defined via the following experiments for $b \in \{0, 1\}$ between the challenger C and the adversary A:

1. *Setup:* C runs $\mathsf{Setup}(1^k)$ and gives EK to A;
2. *Query:* A arbitrarily issues the following oracle queries:
   - *join* oracle query $\mathcal{J}(i)$: C runs $\mathsf{Join}(MSK, i)$ and gives pk_i to A;

- *corruption* oracle query $\mathcal{C}(i)$: C gives sk_i to A;
- *decapsulation* oracle query $\mathcal{D}(i, S, H)$: C runs $\mathsf{Decaps}(EK, sk_i, S, H)$ and gives K to A;

3. *Challenge:* A outputs a set S^* on which it wants to be challenged; C runs $\mathsf{Encaps}(EK, S^*)$ and gets (H^*, K^*), then sets $K = K^*$ if $b = 0$ or picks a random K if $b = 1$, and finally gives (H^*, K) to A;
4. *Query:* A issues further oracle queries as the previous query phase;
5. *Guess:* A outputs a guess b'. The experiment outputs 1 if $b' = b$ and there is no $i^* \in S^*$ for which there has been a $\mathcal{C}(i^*)$ or $\mathcal{D}(i^*, S^*, H^*)$ query. The experiment outputs 0 otherwise.

For any adversary A, we define its *advantage* against BE in an adaptive CCA attack to be the difference between the probability that the above experiment for $b = 0$ outputs 1 and the probability that the experiment for $b = 1$ outputs 1. The scheme is said to be adaptive CCA secure if for any adversary A its advantage against BE in an adaptive CCA attack is negligible in k.

Selective security is defined via similar games with the difference that A commits to the set S^* before the setup phase. For CPA security, A does not get to query the decryption oracle. We sometimes use SCPA, SCCA, ACPA, and ACCA as shorthands referring to selective CPA, selective CCA, adaptive CPA, and adaptive CCA security.

Note that the above definition (which is based on that of [PPS11b][1]) is stronger than that of [BGW05] since they require that the adversary does not make any decryption oracle query with $i \in S^*$ for which $H = H^*$, but we relax the constraint and only require no query with $i \in S^*$ for which $(S, H) = (S^*, H^*)$.

2.2 The BDHE and GBDHE Assumptions

Let us define the two sets of polynomials $P = (p_1, \ldots, p_s)$ and $Q = (q_1, \ldots, q_t)$, with $p_1 = q_1 = 1$, and a polynomial f, where $\forall i, k : p_i, q_k, f \in \mathbb{F}_p[X_1, \ldots, X_n]$. Let us also define $g^P = (g^{p_1}, \ldots, g^{p_s})$. We say that f is independent of (P, Q) if it cannot be written as $f = \sum_{i,j=1}^{s} a_{i,j} p_i p_j + \sum_{k=1}^{t} b_k q_k$ for constants $a_{i,j}$ and b_k.

The *generalized decision bilinear Diffie-Hellman exponent (GBDHE)* problem is defined in [BBG05] as follows: given the input $g^{P(x_1,\ldots,x_n)}$ and $g_{\mathrm{T}}^{Q(x_1,\ldots,x_n)}$ for random choices of $x_1, \ldots, x_n \in \mathbb{F}_p$, decide between $g_{\mathrm{T}}^{f(x_1,\ldots,x_n)}$ and a random $T \in \mathbb{G}_{\mathrm{T}}$. The GBDHE assumption says that it is hard to solve the GBDHE problem if f is independent of (P, Q).

The *decision bilinear Diffie-Hellman exponent* assumption (parameterized by the integer n and denoted by n-BDHE), which is an instance of the GBDHE assumption, says that given the input $g, h, \{g^{\alpha^k}\}_{k \in \{1,\ldots,2n\} \setminus \{n+1\}}$ for random $h \in \mathbb{G}$ and $\alpha \in \mathbb{Z}_p$, it is hard to decide between $e(g, h)^{\alpha^{n+1}}$ and a random $T \in \mathbb{G}_{\mathrm{T}}$.

2.3 Universal One-Way Hash Function

Consider a keyed hash function H. H is called a universal one-way hash function (UOWHF)[2] if there is no efficient adversary winning the following security game. First, the adversary chooses a message and outputs it. Then, the challenger chooses a random key for H and gives it to the adversary. Finally, the adversary outputs a second message and terminates. The adversary wins if the two messages are different, but their hashes under the chosen key are the same. This notion was first proposed in [NY89], and is shown to be strictly weaker than collision resistance [Sim98,RS04]. In fact, one-way functions are shown to be sufficient for UOWHF [Rom90], whereas collision resistant hash functions are only known to be constructed from claw-free permutations [Dam88] or lattice-based assumptions [GGH96].

[1] Note that, in comparison with [PPS11b], we ignore the *Reg* parameter here as it can be regarded as part of *EK*.
[2] UOWHF is also known as target collision resistance (TCR).

3 CCA from Generic Transforms?

In this section we consider the two types of general standard model CPA-to-CCA transforms, namely NY-like and CHK-like, and argue that applying these transforms to the proposed broadcast encryption schemes in the literature does not give us CCA security and constant-size ciphertexts.

NY-like Transforms The Naor-Yung paradigm ([NY90] and [Sah99,DDN00,Lin03]) provides a construction for CCA secure encryption from CPA secure encryption along with non-interactive zero-knowledge proofs. To apply any NY-like transform to a broadcast encryption, one needs to make a NIZK proof of a statement containing the session key K. Such proofs tend to be long and inefficient. Furthermore, all the proposed schemes that have a constant-size ciphertext are pairing-based, and in all these schemes the session key is a member of the target set $\mathbb{G}_{\mathrm{T}}$, but NIZK proofs of statements containing members of $\mathbb{G}_{\mathrm{T}}$ are not known. In particular, Groth-Sahai constructions [GS08] only provide witness indistinguishable proofs for such statements, whereas zero knowledge, and in particular the ability to simulate proofs without knowing a witness, seems to be essential to the security proofs of NY-like constructions.

CHK-like Transforms The Canetti-Halevi-Katz paradigm
([CHK04] and [BK05,Kil06]) provides a construction for CCA secure encryption from CPA secure *identity-based* encryption and an extra authenticating primitive such as signature or message authentication code (MAC). Essential to the paradigm is that any encryption to an identity can be decrypted by the secret key generated for the same identity. However, in the broadcast encryption case, encryptions are made to a set and decryptions are possible by the secret key of any member of that set. Hence, such transforms are not readily applicable to identity-based broadcast encryptions.

4 An Efficient Selective CCA Broadcast Encryption

Let $\mathsf{H}_\kappa : \mathbb{G} \mapsto \mathbb{Z}_p$ be a hash family indexed by κ. We define a broadcast encryption scheme OurBE in the following. We describe the system for (at most) $n-1$ users to be notationally consistent with the original scheme of [BGW05], on which the system is based. The system for n users can be defined accordingly.

- $\mathsf{Setup}(1^k, n-1)$ picks a random generator $g \in \mathbb{G}$, two random quantities $\alpha, \gamma \in \mathbb{Z}_p$, and a random index κ for hash function H, computes $v = g^\gamma$, and outputs $MSK = (\alpha, \gamma)$ and $EK = (g, v, \kappa)$.
- $\mathsf{Join}(MSK, i)$ computes $g_k = g^{(\alpha^k)}$ for $k = i, i+1, n+1-i$, and $n+1+i$, and $d_i = g_i^\gamma$, and outputs $sk_i = d_i$ and $pk_i = (g_i, g_{i+1}, g_{n+1-i}, g_{n+1+i})$. The secret key sk_i is given to the user, and EK is updated by appending pk_i.
- $\mathsf{Encaps}(EK, S)$ picks a random element $t \in \mathbb{Z}_p$ and sets $K = e(g_{n+1}, g)^t$, which can be equivalently computed as $K = e(g_{n+1-i}, g_i)^t$ for any i, computes H as follows, and outputs (H, K).

$$H = \langle g^t, (v \cdot g_1^{\mathsf{H}_\kappa(g^t)} \cdot \prod_{j \in S} g_{n+1-j})^t \rangle.$$

- $\mathsf{Decaps}(EK, sk_i, S, H)$ parses the header as $H = (C_0, C_1)$, checks if the following equation holds:

$$e(C_1, g) = e(v \cdot g_1^{\mathsf{H}_\kappa(C_0)} \cdot \prod_{j \in S} g_{n+1-j}, C_0), \tag{1}$$

  and if it does, then calculates the session key as follows:

$$K = \frac{e(C_1, g_i)}{e(d_i \cdot g_{1+i}^{\mathsf{H}_\kappa(C_0)} \cdot \prod_{j \in S \setminus \{i\}} g_{n+1-j+i},\ C_0)}.$$

In the following we bring a theorem which states that if the hash function H is a universal one-way hash function, then the proposed scheme satisfies selective CCA security under the same assumption as that of the original scheme, namely n-BDHE. Intuitively, the main modification we make in (the encryption algorithm of) the original scheme is the introduction of $g_1^{\mathsf{H}_\kappa(g^t)}$. If this element is not present, as it is in the original scheme, given a header $H = (C_0, C_1)$ corresponding to a key K, one can compute the header (C_0^r, C_1^r) that corresponds to the key K^r, and hence the scheme is malleable. We show that a UOWHF is sufficient to eradicate malleability and get CCA security. This modification is inspired by a similar technique in [BMW05] which, in contrast, was shown to be applicable to an *identity-based* scheme. Here we show that a similar idea is applicable to $\mathsf{BGW_1}$. The proof of the following theorem can be found in Appendix A. In the proof we use the structure of the keys in the scheme to simulate decryption queries.

Theorem 1. *The above scheme is selective CCA secure if the n-BDHE decision problem is hard and H is a universal one-way hash function.*

On Dynamicity Note that the bound on the number of users in OurBE does not prevent the system from being able to handle more than $n-1$ users. That is, as long as the system "jumps over" the users number n and $n+1$ (i.e., after user number $n-1$, the next user is numbered $n+2$), the system can handle polynomially many users more than $n-1$ and remains secure. The security of the scheme with more than $n-1$ users can be proved based on the following assumption: given the input h, and $\{g_k = g^{\alpha^k}\}$ for $k \in \{n+1-m, \ldots, n+1+m\} \setminus \{n+1\}$ for random $g, h \in \mathbb{G}$ and $\alpha \in \mathbb{Z}_p$, it is hard to decide between $e(g_{n+1}, h)$ and a random $T \in \mathbb{G}_{\mathrm{T}}$. It is not hard to see that this assumption is equivalent to the following assumption: given the input g, h, and $\{g_k = g^{\alpha^k}\}$ for $k \in \{1, \ldots, 2m\} \setminus \{m\}$ for random $h \in \mathbb{G}$ and $\alpha \in \mathbb{Z}_p$, it is hard to decide between $e(g_m, h)$ and a random $T \in \mathbb{G}_{\mathrm{T}}$. Here $m \geq n+2$ is the last user number to join. This assumption is comparable to the m-BDHE assumption. In fact, like the BDHE assumption, it is an instance of the GBDHE assumption. In view of this observation, OurBE is a *dynamic* broadcast encryption in the sense that: (1) the system setup and the ciphertext size are independent of the upper bound on the number of users; (2) a new user can join anytime without incurring modification of other user secret keys; and (3) the encryption key is incrementally updated by an operation of O(1) complexity.

Comparison The only broadcast encryption scheme in the literature that provides CCA security with constant-size ciphertexts is $\mathsf{BGW_2}$. It has similar secret and public key sizes as our scheme. However, there are differences in terms of security assumptions and ciphertext size. $\mathsf{BGW_2}$ uses a signature or a message authentication code (MAC) and is proved secure under n-BDHE plus the strong unforgeability (SUF) of the signature or the MAC, whereas OurBE needs n-BDHE plus a universal one-way hash function (UOWHF). In theory, SUF and UOWHF are equivalent (both are equivalent to one-wayness), but in practice, hash functions are generally much more efficient than signatures. In terms of ciphertext size, $\mathsf{BGW_2}$ has a ciphertext whose size is (about) double that of $\mathsf{BGW_1}$'s ciphertext: a $\mathsf{BGW_2}$ ciphertext consists of a $\mathsf{BGW_1}$ ciphertext of two $\mathbb{G}$ elements, plus an element in $\mathbb{Z}_p$ and a signature (or a MAC tag). OurBE has the same ciphertext size as that of $\mathsf{BGW_1}$, i.e., only two $\mathbb{G}$ elements. We summarize this comparison in Table 1. For simplification, we show the total number of elements without the details of the groups to which each element belongs. Note that although pk_i in OurBE includes four group elements, since there are some repeating values the final EK includes the three initial values plus only $2n-1$ extra values of g_i.

5 Inclusive-Exclusive Broadcast Encryption

In this section we show that OurBE can be slightly modified to provide both the broadcast encryption and the revocation functionality simultaneously; that is, we propose a scheme in which the encrypter may choose to determine either a target set S or a revoked set R of users at the time of encryption, without the need to set up two parallel systems. The decryption naturally goes ahead only if the user

Table 1. Comparison of CCA secure schemes with constant-size ciphertexts

Scheme	$\lvert sk_i\rvert$ / $\lvert EK\rvert$	$\lvert H\rvert$	Security	Assumption
[BGW05]	1 / 2n+1	4	SCCA	n-BDHE, SUF
OurBE	1 / 2n+2	2	SCCA	n-BDHE, UOWHF

$|\cdot|$: size in number of elements n: number of users plus one.

is either in S or not in R. In the following we (ab)use the notation "S/R" to indicate "either S or R" as input to the encapsulation and decapsulation algorithms. In practice this can be implemented using the first bit of the input to indicate the inclusive or exclusive mode of operation.

- $\mathsf{Setup}(1^k, n-1)$ picks random $g \in \mathbb{G}$, $\alpha, \gamma \in \mathbb{Z}_p$, and κ for H, computes $v = g^\gamma$, sets $\pi_0 = g^{\alpha(\alpha^n-1)/(\alpha-1)}$, and outputs $MSK = (\alpha, \gamma)$ and $EK = (g, v, \pi_0, \kappa)$.
- $\mathsf{Join}(MSK, i)$ computes g_i, g_{i+1}, g_{n+1-i}, g_{n+1+i}, and $d_i = g_i^\gamma$, sets $\pi_i = \pi_0^{\alpha^i}/g_{n+1}$, and outputs $sk_i = d_i$ and $pk_i = (g_i, g_{i+1}, g_{n+1-i}, g_{n+1+i}, \pi_i)$. Now, sk_i is given to the user, and EK is updated by appending pk_i.
- $\mathsf{Encaps}(EK, S/R)$ picks a random $t \in \mathbb{Z}_p$ and sets $K = e(g_{n+1}, g)^t$, computes H as either of the following accordingly, and outputs (H, K).

$$H = \begin{cases} \langle g^t, (v \cdot g_1^{\mathsf{H}_\kappa(g^t)} \cdot \prod_{j\in S} g_{n+1-j})^t \rangle & \text{if } S \text{ given} \\ \langle g^t, (v \cdot g_1^{\mathsf{H}_\kappa(g^t)} \cdot \pi_0 / \prod_{j\in R} g_{n+1-j})^t \rangle & \text{if } R \text{ given} \end{cases}$$

- $\mathsf{Decaps}(EK, sk_i, S/R, H)$ parses $H = (C_0, C_1)$, checks if the either of the following equation accordingly holds:

$$e(C_1, g) = \begin{cases} e(v \cdot g_1^{\mathsf{H}_\kappa(C_0)} \cdot \prod_{j\in S} g_{n+1-j}, C_0) & \text{if } S \text{ given} \\ e(v \cdot g_1^{\mathsf{H}_\kappa(C_0)} \cdot \pi_0 / \prod_{j\in R} g_{n+1-j}, C_0) & \text{if } R \text{ given} \end{cases}$$

  and if it does, then calculates the session key accordingly as follows:

$$K = \frac{e(C_1, g_i)}{e(d_i \cdot g_{1+i}^{\mathsf{H}_\kappa(C_0)} \cdot \prod_{j\in S\setminus\{i\}} g_{n+1-j+i},\ C_0)}, \quad \text{or}$$

$$K = \frac{e(C_1, g_i)}{e(d_i \cdot g_{1+i}^{\mathsf{H}_\kappa(C_0)} \cdot \pi_i / \prod_{j\in R} g_{n+1-j+i},\ C_0)}.$$

Correctness Let $N = \{1, \ldots, n-1\}$. We have

$$\pi_0 = \prod_{j\in N} g_{n+1-j} \quad \text{and} \quad \pi_i = \prod_{j\in N\setminus\{i\}} g_{n+1-j+i}.$$

Hence in the exclusive mode, for any $i \notin R$ we have:

$$\pi_0 / \prod_{j\in R} g_{n+1-j} = \prod_{j\in N\setminus R} g_{n+1-j} \quad \text{and}$$

$$\pi_i / \prod_{j\in R} g_{n+1-j+i} = \prod_{j\in (N\setminus R)\setminus\{i\}} g_{n+1-j+i}.$$

Hence, if $i \notin R$, the session key the user i calculates in the exclusive mode is effectively the same as the session key it would have calculated if it were decrypting a ciphertext encrypted to $S = N \setminus R$ in the inclusive mode, and therefore the scheme is correct.

Note that the parameters are set in a way that the scheme properly excludes users that join after the time of encryption from inclusive-mode ciphertexts, and includes such users in the exclusive-mode ciphertexts. Unfortunately, the system appears to lose full dynamicity.

Efficiency The scheme enjoys similar desirable efficiency measures as the inclusive-only scheme; that is, the ciphertext and the user secret key sizes are both constant and the public key size is linear in the number of users.

Security A similar security definition to that of broadcast encryption can be defined for such schemes, with the difference that the adversary is now allowed to ask decryption oracle queries for both modes. Naturally exclusive-mode decryption oracle queries $\mathcal{D}(i^*, N \setminus S^*, H^*)$ for $i^* \in S^*$ are also not allowed. It is not hard to see that the security of OurBE translates into the above scheme satisfying this security definition.

6 Achieving Adaptive CCA Security

Since we have a very efficient scheme with asymptotically optimal size secret keys and ciphertexts which is already proved selective CCA secure based on standard assumptions, in this section we try to see how further we can achieve in terms of security by considering reasonable generalizations of some standard assumptions, while retaining the same optimally efficient secret key and ciphertext sizes. We first propose reasonable generalizations of the GBDHE and prove that they hold in the generic group model; then we prove that OurBE can be proved ACCA secure under these assumptions; and finally we compare our scheme to existing adaptive or CCA secure broadcast encryptions.

6.1 The OBDHE Assumption

We consider extending the GBDHE problem assuming that an extra resource is also given: the *Diffie-Hellman computation oracle* $\mathcal{O}^{\mathrm{DH}}_{g,e}$, that takes two inputs $u, v \in \mathbb{G}$ and outputs $w \in \mathbb{G}$ such that $e(u, v) = e(g, w)$. Let us call this the *Oracle BDHE* problem, or OBDHE for short. Formally, we define:

The OBDHE Problem: Given the input $g^{P(x_1,\ldots,x_n)}$ and $g_{\mathrm{T}}^{Q(x_1,\ldots,x_n)}$ for random choices of $x_1, \ldots, x_n \in \mathbb{F}_p$, and access to the $\mathcal{O}^{\mathrm{DH}}_{g,e}$ oracle, decide between $g_{\mathrm{T}}^{f(x_1,\ldots,x_n)}$ and a random $T \in \mathbb{G}_{\mathrm{T}}$.

Note that the GBDHE assumption implies that the only elements (dependent on $x_1, \ldots, x_n$ and) in $\mathbb{G}$ that can be computed are those in the form $g^{\sum a_i p_i}$. Thus, for any $\mathcal{O}^{\mathrm{DH}}_{g,e}$ query (dependent on $x_1, \ldots, x_n$) we can assume $u = g^{\sigma_u}$ and $v = g^{\sigma_v}$, where σ_u and σ_v are polynomials. Then we will have $w = \mathcal{O}^{\mathrm{DH}}_{g,e}(u, v) = g^{\sigma_u \sigma_v}$. Hence, by providing access to $\mathcal{O}^{\mathrm{DH}}_{g,e}$, basically a number of "free multiplications" in the exponent are given. Let us define $p' = \sigma_u \sigma_v$. If we consider q' queries to $\mathcal{O}^{\mathrm{DH}}_{g,e}$, and the output to the i-th query represented as $w_i = g^{p'_i}$, we can define $P' = (p'_1, \ldots, p'_{q'})$. Our extension of the GBDHE assumption says that it is still hard to solve the GBDHE problem if these "free multiplications" in the exponent do not help breaking the independence property. Formally, letting $\|$ denote concatenation, we define:

Assumption 1 (OBDHE) *It is hard to solve the decision (P, Q, f)-OBDHE problem if f is independent of $(P \,\|\, P', Q)$.*

In Appendix B we prove that the assumption holds in the generic group model [Sho97,BBG05]. We prove an upper bound on the success of any generic algorithm trying to solve the OBDHE problem which is negligible if p, the order of $\mathbb{F}_p$ is super-polynomial. Leaving technicalities to the appendix, we prove the following theorem:

Theorem 2. *The OBDHE Assumption holds in the generic group model.*

In fact, our proof is similar to that of [BBG05], suggesting that our assumption is a natural and closely-related extension of GBDHE. It is also worth to note that OBDHE is falsifiable by simply solving the corresponding $(P \parallel P', Q, f)$-GBDHE problem efficiently.

6.2 The GKEA Assumption

We propose the generalized knowledge of exponent assumption (GKEA) as follows and prove that it holds in the generic group model. In the following we use p to denote a polynomial (suppressing the random variables) and $p(x_1, \ldots, x_n)$ to denote the evaluation of p on the input $(x_1, \ldots, x_n)$. Let the tuple $P = (p_1, \ldots, p_s)$ be in $\mathbb{F}_p[X_1, \ldots, X_n]^s$. Let the linear span of P, denoted by $\mathrm{Span}(P)$, be defined as the vector space containing all the polynomials in the form $\sum_{k=1}^{s} a_k p_k$.

Assumption 2 (GKEA) *Let the tuple $P = (p_1, \ldots, p_s)$ be in $\mathbb{F}_p[X_1, \ldots, X_n]^s$, where $p_1 = 1$. Let A be an algorithm that given $g^{P(x_1,\ldots,x_n)}$ for a random $(x_1, \ldots, x_n)$, outputs*

$$(\ (a_k)_{k=1}^{s}, \ h, \ h^{q(x_1,\ldots,x_n)} \ ), \qquad \textit{such that}$$
$$q(x_1, \ldots, x_n) = \sum_{k=1}^{s} a_k p_k(x_1, \ldots, x_n).$$

Consider the subspace of $\mathrm{Span}(P)$ *defined as $V_q = \{r \mid r, rq \in \mathrm{Span}(P)\}$ and let $\{r_i\}_{i=1}^{t}$ be a basis for V_q. Then, there exists an extractor E that given the same input as A outputs*

$$(b_i)_{i=1}^{t}, \qquad \textit{such that} \quad \mathrm{dlog}_g(h) = \sum_{i=1}^{t} b_i r_i(x_1, \ldots, x_n).$$

This assumption basically says that the only way an adversary can produce pairs of the form (h, h^q) is to pick given pairs of the form (h_i, h_i^q) and output $(\prod h_i^{b_i}, \prod (h_i^q)^{b_i})$ for some known values of b_i.

For $P = (1, X)$ and $q(X) = X$, this becomes the original KEA of [Dam92], which basically says that given (g, g^x) the only way an adversary can produce pairs of the form (h, h^x) is to output $(g^b, (g^x)^b)$ for some known value of b. This assumption is referred to KEA1 in [HT98,BP04] and as Diffie-Hellman Knowledge (DHK) in [Den06]. A similar problem is formalized as strong Diffie-Hellman (SDH) in [ABR01].

For $P = (1, X, Y, YX)$ and $q(X, Y) = X$, this becomes the KEA3 assumption of [BP04], which basically says that given (g, g^x, f, f^x) the only way an adversary can produce pairs of the form (h, h^x) is to output $(g^b f^c, (g^x)^b (f^x)^c)$ for some known values of b and c. This assumption is referred to as Extended KEA (XKEA) in [AF07] and as Extended Diffie-Hellman Knowledge (EDHK) in [DP08].

The above two instances of the assumption have already been proved to hold in the generic group model [Den06,AF07,DP08]. In the following we propose a theorem stating the generic assumption and prove it in Appendix C.

Theorem 3. *The GKEA Assumption holds in the generic group model.*

6.3 Adaptive CCA Security

In this section we prove OurBE adaptive CCA secure under our generalized versions of the BDHE and knowledge of exponent assumptions. To prove adaptive CCA security, we basically show that a decryption query by the adversary that contains a valid ciphertext does not increase the (cryptographic) 'knowledge' of the adversary. Also note that since ciphertext validity is publicly verifiable, a decryption

Table 2. Comparison of adaptive or CCA secure broadcast encryption schemes

Scheme		O($\|sk_i\|$)	O($\|H\|$)	Security	Assumption
[DF02]	BE	$\log n$	$r \log \frac{n}{r}$	ACCA1	(IBE)
	BE	$\log^{1+\epsilon} n$	$\frac{r}{\epsilon}$	ACCA1	(HIBE)
[BGW05]	BE	**1**	**1**	SCCA	n-BDHE, SUF
[GW09]	BE	**1**	**1**	ACPA	n-BDHES, PRF, ROM
	BE	**1**	s	ACPA	n-BDHES, PRF
	IBBE	**1**	**1**	ACPA	n-BDHES, PRF, ROM
	IBBE	**1**	$\sqrt{s}$	ACPA	n-BDHES, PRF
[Wat09]	BE	n	**1**	ACPA	dBDH, dLin
[LSW10]	R	**1**	r	ACPA	dBDH, dLin
[PPS11a]	BE	**1**	$r \log \frac{n}{r}$	**ACCA**	DDH
	BE	**1**	r	**ACCA**	DDH
OurBE	BE	**1**	**1**	SCCA	n-BDHE, UOWHF
				ACCA	n-OBDHE, GKEA, UOWHF

O($|\cdot|$): order of size, n, s, r: number of total, targeted, revoked users.

query that contains an invalid ciphertext does not increase the adversary's knowledge either. Hence we basically show that a CCA attack against the system is equivalent to a CPA attack, under the GKEA assumption and the hash function being a UOWHF. Furthermore, the access to $\mathcal{O}^{\mathrm{DH}}_{g,e}$ enables answering adaptive corruption queries.

Formally, we prove adaptive CCA security assuming that the OBDHE and the GKEA assumptions hold and H is a UOWHF. Intuitively, selective CPA security stems from the BDHE assumption underlying the OBDHE assumption along with the hash function being a UOWHF; the Diffie-Hellman oracle enables adaptive security; and the CCA security is achieved from the GKEA assumption along with the hash function being a UOWHF. The following theorem is proved in Appendix D.

Theorem 4. OurBE *is adaptive CCA secure if the OBDHE and the GKEA assumptions hold and* H *is a universal one-way hash function.*

We note that we prove CCA security based on the GKEA assumption, an assumption which is much weaker than the generic model itself (and instances of it are shown to be falsifiable [BP04]), and in fact, proving the equivalence of CPA and CCA security is trivial if the generic group model is used directly, since on a decryption query with a first element g^t, we may assume that t is known.

6.4 Comparison

Since our scheme is the first to achieve adaptive CCA security with constant-size ciphertexts, we compare our scheme with those from the literature that are adaptive CPA or selective CCA secure. We do not consider schemes that are not fully collusion resistant. The schemes in the literature with constant-size ciphertexts include a selective CCA secure scheme from [BGW05], and three adaptive CPA secure schemes from [GW09] and [Wat09]. The schemes in the literature that do not have constant-size ciphertexts include adaptive CPA secure schemes from [DF02], [GW09] (identity-based) and [LSW10] (revocation scheme), and recent adaptive CCA secure schemes from [PPS11a]. Table 2 summarizes our comparison. We consider plain and identity-based (IB) broadcast encryption (BE) and revocation (R) schemes. Among these, schemes from [DF02] and [PPS11a] are generic schemes based on (hierarchical) identity-based encryption ((H)IBE), and encryption schemes (implemented under DDH), respectively. Since (H)IBE can be based on various assumption, we simply use it in parentheses in the table. All other schemes are explicit proposals based on various bilinear Diffie-Hellman assumptions, in some cases plus extra assumptions such as strong unforgeability (SUF) of

signatures, pseudo-random functions (PRF), and the random oracle model (ROM). To accommodate more information, we omit the O notation and write $O(f(n,s,r))$ as $f(n,s,r)$. Comparatively more desirable quantities are highlighted in boldface.

7 Concluding Remarks

We proposed a very efficient broadcast encryption scheme. The sizes of the secret keys and ciphertexts in the scheme are asymptotically optimal, i.e., constant. We showed that the scheme can be proved selective CCA secure assuming BDHE and a universal one-way hash function. Furthermore, we showed that proving adaptive CCA security is possible if we consider extended versions of the GBDHE and knowledge of exponent assumptions. Considering only the standard assumptions, our scheme provides shorter ciphertexts than the only other known CCA secure scheme. Considering the extended assumptions, our scheme is the first scheme to achieve constant size secret keys and ciphertexts and adaptive CCA security at the same time. The problem of designing schemes that achieve such properties under standard assumptions remains open.

Acknowledgments

This work was supported by the French ANR-09-VERS-016 BEST Project. The authors would like to thank the anonymous reviewers of the ACISP 2012 conference and the International Journal of Information Security.

References

ABR01. Michel Abdalla, Mihir Bellare, and Phillip Rogaway. The oracle Diffie-Hellman assumptions and an analysis of DHIES. In David Naccache, editor, *Topics in Cryptology – CT-RSA 2001*, volume 2020 of *Lecture Notes in Computer Science*, pages 143–158. Springer, April 2001.

AF07. Masayuki Abe and Serge Fehr. Perfect NIZK with adaptive soundness. In Salil P. Vadhan, editor, *TCC 2007: 4th Theory of Cryptography Conference*, volume 4392 of *Lecture Notes in Computer Science*, pages 118–136. Springer, February 2007.

BBG05. Dan Boneh, Xavier Boyen, and Eu-Jin Goh. Hierarchical identity based encryption with constant size ciphertext. In Ronald Cramer, editor, *Advances in Cryptology – EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 440–456. Springer, May 2005.

Ber70. Elwyn R. Berlekamp. Factoring polynomials over large finite fields. *Mathematics of Computation*, 24(111):pp. 713–735, 1970.

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

BK05. Dan Boneh and Jonathan Katz. Improved efficiency for CCA-secure cryptosystems built using identity-based encryption. In Alfred Menezes, editor, *Topics in Cryptology – CT-RSA 2005*, volume 3376 of *Lecture Notes in Computer Science*, pages 87–103. Springer, February 2005.

BMW05. Xavier Boyen, Qixiang Mei, and Brent Waters. Direct chosen ciphertext security from identity-based techniques. In Vijayalakshmi Atluri, Catherine Meadows, and Ari Juels, editors, *ACM CCS 05: 12th Conference on Computer and Communications Security*, pages 320–329. ACM Press, November 2005.

BP04. Mihir Bellare and Adriana Palacio. The knowledge-of-exponent assumptions and 3-round zero-knowledge protocols. In Matthew Franklin, editor, *Advances in Cryptology – CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 273–289. Springer, August 2004.

BR93. Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In V. Ashby, editor, *ACM CCS 93: 1st Conference on Computer and Communications Security*, pages 62–73. ACM Press, November 1993.

CHK04. Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology – EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 207–222. Springer, May 2004.

CS03. Ronald Cramer and Victor Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM Journal on Computing*, 33(1):167–226, 2003.

Dam88. Ivan Damgård. Collision free hash functions and public key signature schemes. In David Chaum and Wyn L. Price, editors, *Advances in Cryptology – EUROCRYPT'87*, volume 304 of *Lecture Notes in Computer Science*, pages 203–216. Springer, April 1988.

Dam92. Ivan Damgård. Towards practical public key systems secure against chosen ciphertext attacks. In Joan Feigenbaum, editor, *Advances in Cryptology – CRYPTO'91*, volume 576 of *Lecture Notes in Computer Science*, pages 445–456. Springer, August 1992.

DDN00. Danny Dolev, Cynthia Dwork, and Moni Naor. Nonmalleable cryptography. *SIAM Journal on Computing*, 30(2):391–437, 2000.

Del07. Cécile Delerablée. Identity-based broadcast encryption with constant size ciphertexts and private keys. In Kaoru Kurosawa, editor, *Advances in Cryptology – ASIACRYPT 2007*, volume 4833 of *Lecture Notes in Computer Science*, pages 200–215. Springer, December 2007.

Den06. Alexander W. Dent. The hardness of the dhk problem in the generic group model. Cryptology ePrint Archive, Report 2006/156, 2006. `http://eprint.iacr.org/2006/156`.

DF02. Yevgeniy Dodis and Nelly Fazio. Public-key broadcast encryption for stateless receivers. In *ACM Digital Rights Management—DRM '02*, pages 61–80, Heidelberg, 2002. Springer. LNCS 2696.

DF03. Yevgeniy Dodis and Nelly Fazio. Public key trace and revoke scheme secure against adaptive chosen ciphertext attack. In Yvo Desmedt, editor, *PKC 2003: 6th International Workshop on Theory and Practice in Public Key Cryptography*, volume 2567 of *Lecture Notes in Computer Science*, pages 100–115. Springer, January 2003.

DP08. Yvo Desmedt and Duong Hieu Phan. A CCA secure hybrid Damgård's ElGamal encryption. In Joonsang Baek, Feng Bao, Kefei Chen, and Xuejia Lai, editors, *ProvSec 2008: 2nd International Conference on Provable Security*, volume 5324 of *Lecture Notes in Computer Science*, pages 68–82. Springer, October / November 2008.

DPP07. Cécile Delerablée, Pascal Paillier, and David Pointcheval. Fully collusion secure dynamic broadcast encryption with constant-size ciphertexts or decryption keys. In Tsuyoshi Takagi, Tatsuaki Okamoto, Eiji Okamoto, and Takeshi Okamoto, editors, *PAIRING 2007: 1st International Conference on Pairing-based Cryptography*, volume 4575 of *Lecture Notes in Computer Science*, pages 39–59. Springer, July 2007.

FN94. Amos Fiat and Moni Naor. Broadcast encryption. In Douglas R. Stinson, editor, *Advances in Cryptology – CRYPTO'93*, volume 773 of *Lecture Notes in Computer Science*, pages 480–491. Springer, August 1994.

GGH96. Oded Goldreich, Shafi Goldwasser, and Shai Halevi. Collision-free hashing from lattice problems. Cryptology ePrint Archive, Report 1996/009, 1996. `http://eprint.iacr.org/1996/009`.

GS08. Jens Groth and Amit Sahai. Efficient non-interactive proof systems for bilinear groups. In Nigel P. Smart, editor, *Advances in Cryptology – EUROCRYPT 2008*, volume 4965 of *Lecture Notes in Computer Science*, pages 415–432. Springer, April 2008.

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

HT98. Satoshi Hada and Toshiaki Tanaka. On the existence of 3-round zero-knowledge protocols. In Hugo Krawczyk, editor, *Advances in Cryptology – CRYPTO'98*, volume 1462 of *Lecture Notes in Computer Science*, pages 408–423. Springer, August 1998.

Kil06. Eike Kiltz. Chosen-ciphertext security from tag-based encryption. In Shai Halevi and Tal Rabin, editors, *TCC 2006: 3rd Theory of Cryptography Conference*, volume 3876 of *Lecture Notes in Computer Science*, pages 581–600. Springer, March 2006.

Lin03. Yehuda Lindell. A simpler construction of cca2-secure public-key encryption under general assumptions. In Eli Biham, editor, *Advances in Cryptology – EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 241–254. Springer, May 2003.

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

Nie02. Jesper Buus Nielsen. Separating random oracle proofs from complexity theoretic proofs: The non-committing encryption case. In Moti Yung, editor, *Advances in Cryptology – CRYPTO 2002*, volume 2442 of *Lecture Notes in Computer Science*, pages 111–126. Springer, August 2002.

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

NY89. Moni Naor and Moti Yung. Universal one-way hash functions and their cryptographic applications. In *21st Annual ACM Symposium on Theory of Computing*, pages 33–43. ACM Press, May 1989.

NY90. Moni Naor and Moti Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *22nd Annual ACM Symposium on Theory of Computing*. ACM Press, May 1990.

PPS11a. Duong Hieu Phan, David Pointcheval, and Mario Strefler. Adaptively secure broadcast encryption with forward secrecy. Cryptology ePrint Archive, Report 2011/463, 2011. `http://eprint.iacr.org/2011/463`.

PPS11b. Duong Hieu Phan, David Pointcheval, and Mario Strefler. Security notions for broadcast encryption. In Javier Lopez and Gene Tsudik, editors, *ACNS'11*, volume 6715 of *Lecture Notes in Computer Science*, pages 377–394, 2011.

PPSS12. Duong Hieu Phan, David Pointcheval, Siamak F. Shahandashti, and Mario Strefler. Adaptive cca broadcast encryption with constant-size secret keys and ciphertexts. In Willy Susilo, Yi Mu, and Jennifer Seberry, editors, *ACISP*, volume 7372 of *Lecture Notes in Computer Science*, pages 308–321. Springer, 2012.

Rom90. John Rompel. One-way functions are necessary and sufficient for secure signatures. In *22nd Annual ACM Symposium on Theory of Computing*, pages 387–394. ACM Press, May 1990.

RS04. Phillip Rogaway and Thomas Shrimpton. Cryptographic hash-function basics: Definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance. In Bimal K. Roy and Willi Meier, editors, *Fast Software Encryption – FSE 2004*, volume 3017 of *Lecture Notes in Computer Science*, pages 371–388. Springer, February 2004.

Sah99. Amit Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *40th Annual Symposium on Foundations of Computer Science*, pages 543–553. IEEE Computer Society Press, October 1999.

Sho90. Victor Shoup. On the deterministic complexity of factoring polynomials over finite fields. *Information Processing Letters*, 33(5):261 – 267, 1990.

Sho97. Victor Shoup. Lower bounds for discrete logarithms and related problems. In Walter Fumy, editor, *Advances in Cryptology – EUROCRYPT'97*, volume 1233 of *Lecture Notes in Computer Science*, pages 256–266. Springer, May 1997.

Sim98. Daniel R. Simon. Finding collisions on a one-way street: Can secure hash functions be based on general assumptions? In Kaisa Nyberg, editor, *Advances in Cryptology – EUROCRYPT'98*, volume 1403 of *Lecture Notes in Computer Science*, pages 334–345. Springer, May / June 1998.

Wat09. Brent Waters. Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In Shai Halevi, editor, *Advances in Cryptology – CRYPTO 2009*, volume 5677 of *Lecture Notes in Computer Science*, pages 619–636. Springer, August 2009.

A Proof of Theorem 1

Proof. Suppose there exist a selective CCA adversary A that is able to distinguish the above scheme's keys from random elements. We construct an algorithm B that either outputs a collision for a given key κ or solves the n-BDHE decision problem.

Let B be given an n-BDHE challenge $(g, h, \{g_i\}_{i\in\{1,\ldots,2n\}\setminus\{n+1\}}, T)$ and has to decide whether $T = e(g_{n+1}, h)$ or it is random. B runs A and receives a set S^* of honest users on which it wishes to be challenged. As a UOWHF adversary, B also gives out h as the first message on which it wishes to be challenged and receives a key κ for the hash function. B chooses a random $\beta \in \mathbb{Z}_p$, calculates v as follows, and gives $EK = (g, v, \kappa)$ to A.

$$v = g^{\beta} \cdot g_1^{-\mathsf{H}_\kappa(h)} \cdot \prod_{j\in S^*} g_{n+1-j}^{-1}. \tag{2}$$

On any join query for user i made by the adversary, B gives $pk_i = (g_i, g_{n+1-i}, g_{n+1+i})$ to A.

On any private key query for user i made by A (note that $i \notin S^*$), B calculates the private key as follows and gives it to A.

$$d_i = g_i^{\beta} \cdot g_{1+i}^{-\mathsf{H}_\kappa(h)} \cdot \prod_{j\in S^*} g_{n+1-j+i}^{-1}.$$

Note that d_i is properly simulated since we have $d_i = v^{\alpha^i}$.

On a decryption query $(i, S, (C_0, C_1))$ by A (note that $S \subset S^*$ and $i \in S$), B first checks the validity of the ciphertext using Equation 1. If the ciphertext is valid then it checks whether $\mathsf{H}_\kappa(h) = \mathsf{H}_\kappa(C_0)$ which in case of validity provides a collision for the hash function H_κ and hence B can output C_0 as the second message and break the UOWHF property.

If Equation 1 holds and $\mathsf{H}_\kappa(h) \neq \mathsf{H}_\kappa(C_0)$, then let $\delta = \mathsf{H}_\kappa(C_0) - \mathsf{H}_\kappa(h)$. B calculates the key as follows:

$$K = \frac{e(C_1, g \cdot g_n^{1/\delta})}{e(g^{\beta} \cdot g_n^{\beta/\delta} \cdot g_1^{\delta} \cdot \prod_{j\in S^*\setminus S} (g_{n+1-j} \cdot g_{2n+1-j}^{1/\delta})^{-1}, C_0)}.$$

Now since Equation 1 holds, the ciphertext is in the form

$$(g^t, (v \cdot g_1^{\mathsf{H}_\kappa(g^t)} \cdot \prod_{j\in S} g_{n+1-j})^t)$$

for some (unknown) t. Hence, the above calculated K will be as follows:

$$
\begin{aligned}
K &= \frac{e((v \cdot g_1^{\mathsf{H}_\kappa(g^t)} \cdot \prod_{j\in S} g_{n+1-j})^t, g \cdot g_n^{1/\delta})}{e(g^\beta \cdot g_n^{\beta/\delta} \cdot g_1^\delta \cdot \prod_{j\in S^*\setminus S} (g_{n+1-j} \cdot g_{2n+1-j}^{1/\delta})^{-1}, g^t)} \\
&= \left(\frac{e(g^\beta \cdot g_1^\delta \cdot \prod_{j\in S^*\setminus S} g_{n+1-j}^{-1}, g \cdot g_n^{1/\delta})}{e(g^\beta \cdot g_n^{\beta/\delta} \cdot g_1^\delta \cdot \prod_{j\in S^*\setminus S} (g_{n+1-j} \cdot g_{2n+1-j}^{1/\delta})^{-1}, g)} \right)^t \\
&= \left(\frac{e(g^\beta \cdot g_1^\delta \cdot \prod_{j\in S^*\setminus S} g_{n+1-j}^{-1}, g^{1+\alpha^n/\delta})}{e(g^{\beta(1+\alpha^n/\delta)} \cdot g_1^\delta \cdot \prod_{j\in S^*\setminus S} g_{n+1-j}^{-(1+\alpha^n/\delta)}, g)} \right)^t \\
&= \left(\frac{e(g_1^\delta, g^{1+\alpha^n/\delta})}{e(g_1^\delta, g)} \right)^t = e(g_1, g)^{\alpha^n t} = e(g_{n+1}, g)^t
\end{aligned}
$$

and hence it is properly simulated. In the above, we have substituted v from Equation 2 and used the fact that $\forall k : g_{n+k} = g_k^{\alpha^n}$.

At some point, A declares that it is ready to receive the challenge. B calculates the challenge ciphertext as $C = (h, h^\beta)$ and gives C along with $K = T$ to A. First, note that from Equation 2 we have

$$v \cdot g_1^{\mathsf{H}_\kappa(h)} \cdot \prod_{j\in S^*} g_{n+1-j} = g^\beta,$$

and hence C is a valid ciphertext satisfying Equation 1. Furthermore, assuming that $h = g^t$ for some t, we have

$$h^\beta = (g^\beta)^t = (v \cdot g_1^{\mathsf{H}_\kappa(h)} \cdot \prod_{j\in S^*} g_{n+1-j})^t,$$

which means that if $T = e(g_{n+1}, h) = e(g_{n+1}, g)^t$, then K is the key corresponding to the ciphertext C, and if T is random, then K is a random key.

In the second phase of the attack, B answers A's queries as in the first phase.

At the end, A outputs its guess b. B outputs b as its decision for the n-BDHE challenge. Based on the above discussion, if A is successful in its CCA attack, then either B is able to compute a collision for H_κ and win the UOWHF game, or it is able to solve the n-BDHE decision problem successfully. □

B Proof of the OBDHE Assumption

In this section, we prove Theorem 2. Let d_P, $d_{P'}$, d_Q, and d_f be respectively the maximum degrees of the polynomials in P, P', Q, and f. We prove the following upper bound in the generic bilinear group model. We consider two random encodings $\xi, \zeta : \mathbb{Z}_p^+ \mapsto \{0,1\}^m$ and write $\mathbb{G} = \{\xi(x) | x \in \mathbb{Z}_p^+\}$ and $\mathbb{G}_\mathrm{T} = \{\zeta(x) | x \in \mathbb{Z}_p^+\}$. The following theorem is a sufficient condition for Theorem 2.

Theorem 5. *For P, Q, P' f, ξ, ζ, $\mathbb{G}$, $\mathbb{G}_\mathrm{T}$ defined above, let $|P| = s$, $|Q| = t$, and $\ell = s + t$. Let $d = \max(2d_P, d_Q, d_f)$. If f is independent of $(P \parallel P', Q)$, then for any A making a total of at most q queries to the oracles computing the group operations and the bilinear pairing, and at most q' queries*

to the $\mathcal{O}^{\mathrm{DH}}_{g,e}$ oracle, we have:

$$\left|\Pr\left[\mathsf{A}\begin{pmatrix} p, \xi(P(x_1,\ldots,x_n)), \\ \zeta(Q(x_1,\ldots,x_n)), \\ \zeta(t_0), \zeta(t_1); \mathcal{O}^{\mathrm{DH}}_{g,e}(\cdot,\cdot)\end{pmatrix} = b : \begin{matrix} x_1,\ldots,x_n, y \xleftarrow{\mathrm{R}} \mathbb{F}_p, \\ b \xleftarrow{\mathrm{R}} \{0,1\}, \\ t_b \leftarrow f(x_1,\ldots,x_n), \\ t_{1-b} \leftarrow y \end{matrix}\right] - \tfrac{1}{2}\right| \leq \frac{(q+q'+\ell+2)^2 \cdot \max(2d_{P'}, d)}{2p}$$

Proof. Assume that we are given the algorithm A. Consider an algorithm B that interacts with A as follows. B maintains two lists of pairs:

$$L = \{(p_i, \xi_i) : i = 1, \ldots, \tau_0\} \quad \text{and} \quad L_{\mathrm{T}} = \{(q_i, \zeta_i) : i = 1, \ldots, \tau_1\},$$

such that at step τ of its interaction with A: $\tau_0 + \tau_1 = \tau + \ell + 2$. Here, $p_i \in \mathbb{F}_p[X_1, \ldots, X_n]$, $q_i \in \mathbb{F}_p[X_1, \ldots, X_n, Y_0, Y_1]$, and $\xi_i, \zeta_i \in \{0,1\}^m$.

B also maintains a counter τ', initialized at zero, to count the number of $\mathcal{O}^{\mathrm{DH}}_{g,e}$ oracle queries, and a list of polynomials:

$$P' = \{p'_i : i = 1, \ldots, \tau'\}$$

to store the polynomial output of the $\mathcal{O}^{\mathrm{DH}}_{g,e}$ oracle queries.

At step $\tau = 0$, B initializes the lists by setting $p_1, \ldots, p_s$ in L equal to the polynomials in P, $q_1, \ldots, q_t$ in L_{T} equal to the polynomials in Q, $q_{t+1} = Y_0$, and $q_{t+2} = Y_1$. It also chooses $\ell + 2$ random strings in $\{0,1\}^m$ and initializes $\{\xi_i\}_{i=1}^{s}$ and $\{\zeta_i\}_{i=1}^{t+2}$.

B then runs A under the input p, $\{\xi_i\}_{i=1}^{s}$, $\{\zeta_i\}_{i=1}^{t}$, ζ_{t+1}, and ζ_{t+2}. B answers A's oracle queries as follows. We are assuming that A's queries can only be strings obtained from B since B can, by increasing m, make the strings in $\mathbb{G}$ and $\mathbb{G}_{\mathrm{T}}$ arbitrarily hard to guess.

Group operations: For a $\mathbb{G}$ operation query (ξ_i, ξ_j), B calculates $p_{\tau_0+1} \leftarrow p_i \pm p_j$ depending on whether multiplication or division is requested. If $p_{\tau_0+1} = p_l$ for some $l \leq \tau_0$, then B sets $\xi_{\tau_0+1} \leftarrow \xi_l$; otherwise it sets ξ_{τ_0+1} equal to a new random string different from all the previous ξ_i. Then it appends the new pair $(p_{\tau_0+1}, \xi_{\tau_0+1})$ to L, replies to A's query with ξ_{τ_0+1}, and finally increments the counter τ_0. $\mathbb{G}_{\mathrm{T}}$ operation queries are dealt with analogously by updating the list L_{T} and counter τ_1.

Bilinear pairings: For a pairing query of the form (ξ_i, ξ_j), B calculates $q_{\tau_1+1} \leftarrow p_i \cdot p_j$. If $q_{\tau_1+1} = q_l$ for some $l \leq \tau_1$, then B sets $\zeta_{\tau_1+1} \leftarrow \zeta_l$; otherwise it sets ζ_{τ_1+1} equal to a new random string different from all the previous ζ_i. Then it appends the new pair $(q_{\tau_1+1}, \zeta_{\tau_1+1})$ to L_{T}, replies to A's query with ζ_{τ_1+1}, and finally increments the counter τ_1.

$\mathcal{O}^{\mathrm{DH}}_{g,e}$ queries: For a $\mathcal{O}^{\mathrm{DH}}_{g,e}$ query (ξ_i, ξ_j), B calculates $p_{\tau_0+1} \leftarrow p_i \cdot p_j$. If $p_{\tau_0+1} = p_l$ for some $l \leq \tau_0$, then B sets $\xi_{\tau_0+1} \leftarrow \xi_l$; otherwise it sets ξ_{τ_0+1} equal to a new random string different from all the previous ξ_i. B also sets $p'_{\tau'+1} \leftarrow p_{\tau_0+1}$, appends $p'_{\tau'+1}$ to P', and increments the counter τ'. Then it appends the new pair $(p_{\tau_0+1}, \xi_{\tau_0+1})$ to L, replies to A's query with ξ_{τ_0+1}, and finally increments the counter τ_0.

A terminates after at most $q + q'$ queries and returns a guess b'.

Now B chooses $x_1, \ldots, x_n, y \xleftarrow{\mathrm{R}} \mathbb{F}_p$ and $b \xleftarrow{\mathrm{R}} \{0,1\}$, and sets $y_b \leftarrow f(x_1, \ldots, x_n)$ and $y_{1-b} \leftarrow y$. Setting $X_i = x_i$ for all $i = 1, \ldots, n$, $Y_0 = y_0$, and $Y_1 = y_1$, we see that B's interaction provides a perfect simulation for A as long as the chosen random values for the random variables do not result in any equality of the values of the intermediate different polynomials. In other words, the simulation is perfect unless for some i and j we have one of the following:

1. $p_i(x_1, \ldots, x_n) = p_j(x_1, \ldots, x_n)$, yet the polynomials p_i and p_j are not equal, or
2. $q_i(x_1, \ldots, x_n, y_0, y_1) = q_j(x_1, \ldots, x_n, y_0, y_1)$, yet the polynomials q_i and q_j are not equal.

Let Fail be the event that one of the above conditions holds. We bound the probability of this event.

First, if we set $Y_b = f(X_1, \ldots, X_n)$, this does not raise the probability that Fail happens. This is because the above substitution does not create any new equalities between polynomials q_i and q_j. In general, $q_i - q_j$ is in the form

$$\sum_{k=1}^{s}\sum_{l=1}^{s} a_{k,l}p_k p_l + \sum_{k=1}^{s}\sum_{l=1}^{q'} a'_{k,l}p_k p'_l + \sum_{k=1}^{q'}\sum_{l=1}^{q'} a''_{k,l}p'_k p'_l + \sum_{u=1}^{t} b_u q_u + cY_0 + dY_1.$$

Let us define

$$P^* = P \parallel P' = (p^*_1, \ldots, p^*_{s+q'}) = (p_1, \ldots, p_s, p'_1, \ldots, p'_{q'}).$$

Now we can write $q_i - q_j$ in the form

$$\sum_{k=1}^{s+q'}\sum_{l=1}^{s+q'} a_{k,l}p^*_k p^*_l + \sum_{u=1}^{t} b_u q_u + cY_0 + dY_1.$$

Hence assuming that the substitution $Y_b = f(X_1, \ldots, X_n)$, does create a new equality, then $q_i - q_j$, which is in the above form, is a non-zero polynomial, yet setting $Y_b = f(X_1, \ldots, X_n)$ makes it zero. Thus, f must be dependent on $(P \parallel P', Q)$, which is a contradiction.

Now with the substitution $Y_b = f(X_1, \ldots, X_n)$, our polynomials are only in $X_1, \ldots, X_n$, and Y_{1-b}. The maximum degree of any polynomial in the form $p_i - p_j$ or $q_i - q_j$ is $\max(2d_P, 2d_{P'}, d_Q, d_f) = \max(2d_{P'}, d)$. Hence, for each pair (i, j), the probability that a random assignment of the random variables is a root of one of the above polynomials is at most $\max(2d_{P'}, d)/p$. Since there are at most $2\binom{q+q'+\ell+2}{2}$ pairs of (p_i, p_j) and (q_i, q_j) in total, we have

$$\begin{aligned}\Pr[\text{Fail}] &\le \binom{q+q'+\ell+2}{2}\frac{2\max(2d_{P'}, d)}{p} \\ &\le \frac{(q+q'+\ell+2)^2\max(2d_{P'}, d)}{p}.\end{aligned}$$

Now we would like to bound A's success probability, i.e., $|\Pr[b = b'] - \frac{1}{2}|$. We know that

$$\Pr[b = b'] = \Pr[b = b'|\text{Fail}] \cdot \Pr[\text{Fail}] + \Pr[b = b'|\neg\text{Fail}] \cdot \Pr[\neg\text{Fail}].$$

If Fail does not happen, then B's simulation is perfect. In this case, since b is chosen after the simulation ends, $\Pr[b = b'|\neg\text{Fail}] = \frac{1}{2}$. Substituting this and $\Pr[\neg\text{Fail}] = 1 - \Pr[\text{Fail}]$ in the above equation, we get the following after rearrangement:

$$\Pr[b = b'] - \frac{1}{2} = (\Pr[b = b'|\text{Fail}] - \frac{1}{2}) \cdot \Pr[\text{Fail}].$$

Hence we have

$$|\Pr[b = b'] - \frac{1}{2}| = |\Pr[b = b'|\text{Fail}] - \frac{1}{2}| \cdot \Pr[\text{Fail}] \le \frac{1}{2}\Pr[\text{Fail}],$$

which gives us the claimed bound and finishes the proof. □

C Proof of Theorem 3

Proof. Let d_P be the maximum degree of the polynomials in P. We consider a random encoding $\xi : \mathbb{Z}_p^+ \mapsto \{0,1\}^m$ and write $\mathbb{G} = \{\xi(x)|x \in \mathbb{Z}_p^+\}$.

Given an algorithm A we construct the extractor E as follows. E maintains a list L of pairs (p_i, ξ_i), initialized with pairs containing the elements of P and random strings, respectively as the first and second elements.

E runs A on input $(\xi_i)_{i=1}^s$. Any group operation query (ξ_i, ξ_j) is responded by computing $p_i + p_j$ and checking if the resulting polynomial already exists in the list. If it does, E returns the corresponding encoding, and if not, it chooses a new random string as the encoding to be returned, and adds $p_i + p_j$ and the encoding to the list L.

When A terminates and returns (ξ_i, ξ_j) as its output, E finds the corresponding polynomial pair (p_i, p_j). If $p_j \neq p_i q$, E outputs $\perp$. Otherwise, let $\{r_i\}_{i=1}^t$ be defined as above. E decomposes p_i as a linear combination of $\{r_i\}_{i=1}^t$, that is, it finds coefficients $(b_i)_{i=1}^t$ such that $p_i = \sum_{i=1}^t b_i r_i$, and outputs $(b_i)_{i=1}^t$.

Assume that A asks σ queries. E's list contains $s + \sigma$ pairs at the end of the execution of A. All the polynomials in this list are in $\mathrm{Span}(P)$. Since both p_i and p_j are in $\mathrm{Span}(P)$, if $p_j = p_i q$, then $p_i \in V_q$, and hence p_i can be written as a linear combination of $\{r_i\}_{i=1}^t$. Furthermore, the discrete logarithm of A's first input ξ_i is equal to $p_i(x_1, \ldots, x_n)$, which in turn equals $\sum_{i=1}^t b_i r_i(x_1, \ldots, x_n)$. Therefore, E succeeds if its simulation of A's environment is perfect and $p_j = p_i q$.

Note that if A's environment is simulated perfectly, then it outputs a pair for which we have $p_j(x_1, \ldots, x_n) = p_i(x_1, \ldots, x_n) q(x_1, \ldots, x_n)$, but not necessarily $p_j = p_i q$.

Let Fail be the event that E fails. Based on the above discussion, E fails if either it fails to simulate A's environment perfectly or if $p_j \neq p_i q$ but $p_j(x_1, \ldots, x_n) = p_i(x_1, \ldots, x_n) q(x_1, \ldots, x_n)$. E's simulation of the environment for A is perfect unless a set of random values $(x_1, \ldots, x_n)$ result in some equality of the values of the different polynomials in L. Hence, if we add $p_i q$ as the polynomial number $s + \sigma + 1$ to the list L, E's overall probability of failure is bounded by the probability that a set of random values $(x_1, \ldots, x_n)$ result in some equality of the values of the different polynomials in the augmented list of $s + \sigma + 1$ polynomials. Hence we have:

$$\Pr[\text{Fail}] \leq \binom{s+\sigma+1}{2} \frac{d_P}{p} \leq \frac{(s+\sigma+1)^2 d_P}{p},$$

and the proof is complete. □

The above proof is in the plain generic group model. It is easy to extend the proof to the bilinear generic group model. Furhtermore, one can see that the proof still works (with some natural modifications) in the model where the adversary is allowed to query the oracles on any encoding, rather than only those it has received before (either as input or as responses to previous oracle queries).

Another point to note is that, in the bilinear group model, any input to the adversary in the target group can be disregarded and hence does not change the assumption.

D Proof of Theorem 4

Proof. We make our proof in two stages.

Stage 1: First, we prove that if H is a UOWHF, then the following specific assumption is an instance of the OBDHE assumption as per our definition in Section 6.1: let $\mathcal{O}_{g,e}^{\mathrm{DH}}$ be an oracle that given (x_1, x_2) outputs y s.t. $e(x_1, x_2) = e(g, y)$. Given the following quantities:

$$g,\ h,\ \{g_k = g^{\alpha^k}\}_{k \in \{1,\ldots,2n\} \setminus \{n+1\}},\ v,$$

and oracle access to $\mathcal{O}_{g,e}^{\mathrm{DH}}$, it is hard to distinguish $e(g^{\alpha^{n+1}}, h)$ from a random value if the queries to $\mathcal{O}_{g,e}^{\mathrm{DH}}$ are restricted to the following, where $C \cap S = \varnothing$:

(1) $|C|$ queries $\{\mathcal{O}_{g,e}^{\mathrm{DH}}(g_k, v)\}_{k \in C}$, and
(2) one query $\mathcal{O}_{g,e}^{\mathrm{DH}}(w, h)$, where $w = v g_1^{\mathsf{H}_\kappa(h)} \prod_{j \in S} g_{n+1-j}$.

Consider the hash function $\mathsf{H}_\kappa : \mathbb{G} \mapsto \mathbb{Z}_p$ and define the function $\mu(h) = h^{\mathsf{H}_\kappa(h)}$. In the generic group model, the input to H_κ is an encoding representing h, which is considered to be an encoding

that may be chosen *independently* of h. Therefore, we may assume $\mathsf{H}_\kappa(h)$ independent of h. Of course this is true only if the sole way to calculate $\mu(h)$ is through computing $\mathsf{H}_\kappa(h)$ first and then raising h to the power of the hash output. Otherwise, if $\mu(h)$ cannot be computed through group operations, without computing $\mathsf{H}_\kappa(h)$ separately, then the encoding of h cannot be chosen independently of h. For a "good" hash function we may assume that $\mu(h)$ cannot be computed through group operations, without computing $\mathsf{H}_\kappa(h)$ separately.

To be more precise, consider Theorem 5 and its presented proof in Appendix B. Assume that P also includes an extra element which is a multiplication of a polynomial and the function $\eta(y) = \mathsf{H}_\kappa(g^y)$. Now, if the encoding of $h = g^y$ is chosen independently of h, the proof will still work, i.e., Pr[FAIL] can be shown to be upper-bounded by a negligible bound, unless for some considerable portion of possible y's we have $\rho_1(y)\eta^2(y) + \rho_2(y)\eta(y) + \rho_3(y) = 0$, where ρ_1, ρ_2, and ρ_3 are polynomials of degree at most $\max(2d_{P'}, d)$. This condition implies that $\eta(y)$ can be calculated for some considerable portion of possible y's by solving the above equation.

Formally, let us define a *δ-good* hash family as follows: We say a hash family $\mathsf{H}_\kappa : \mathbb{G} \mapsto \mathbb{Z}_p$ indexed by κ is *δ-good* if for a random κ there does not exist polynomials ρ_1, ρ_2, and ρ_3 of degree at most δ such that for a non-negligible portion of possible y's we have: $\rho_1(y)\ \mathsf{H}_\kappa^2(g^y) + \rho_2(y)\ \mathsf{H}_\kappa(g^y) + \rho_3(y) = 0$. Now since $\max(2d_{P'}, d) = 4n$, we conclude that if H is at least $4n$-good, then its output can be considered independent of the encoding of its input, and hence we may treat it as a constant.

Now assume that for a given random κ and Y, we wish to find a pre-image X, such that $\mathsf{H}_\kappa(X) = Y$. Assume $X = g^x$. If H is not a δ-good hash family, for a random κ there exist polynomials ρ_1, ρ_2, and ρ_3 of degree at most δ such that with a non-negligible probability: $\rho_1(x)\ Y^2 + \rho_2(x)\ Y + \rho_3(x) = 0$. This is a polynomial of order at most δ, and its roots can be found in time which is polynomial in δ and $\log p$ [Ber70,Sho90]. For each root x, one can check whether $\mathsf{H}_\kappa(g^x) = Y$ and find the pre-image X with at most δ checks. Hence, if H is not a δ-good hash family, then it is not a pre-image resistant (a.k.a. one-way) hash function. Since UOWHF implies pre-image resistance, we have the following lemma:

Lemma 6. *Let $\mathsf{H}_\kappa : \mathbb{G} \mapsto \mathbb{Z}_p$ be hash function for which p is super-polynomial in k. If H is a universal one-way hash function, then it is δ-good (as per our definition above) for all δ polynomial in k.*

Hence, if H is a UOWHF, then the following claim proves that the specific assumption above is an OBDHE assumption as per our definition in Section 6.1, in which the output of H is treated as a constant. Note that alternatively one may make the stronger assumption that H is modeled as a non-programmable random oracle [BR93,Nie02]. Also note that since the system is defined for $n-1$ users, S and C are subsets of $\{1, \ldots, n-1\}$.

Claim. For the following polynomials and $S, C \subseteq \{1, \ldots, n-1\}$, and for any constant c, f is independent of $(P \parallel P', Q)$ if $C \cap S = \varnothing$.

$$P = (\ 1,\ y,\ \{x^k\}_{k\in\{1,\ldots,2n\}\setminus\{n+1\}},\ z,\ \eta,\ zy + \mathrm{c}xy + y\sum_{j\in S} x^{n+1-j}\ ),$$
$$P' = \{zx^i\}_{i\in C}, \qquad Q = (1), \qquad \text{and} \qquad f = yx^{n+1}.$$

Proof. We have at most one multiplication of polynomials at our disposal. Let us define

$$P_x = \{x^k\}_{k\in\{1,\ldots,2n\}\setminus\{n+1\}}, \quad P_{zx} = \{zx^i\}_{i\in C}, \quad \text{and}$$
$$P_{yx} = P_{yz} = zy + \mathrm{c}xy + y\sum_{j\in S} x^{n+1-j}.$$

To make $f = yx^{n+1}$, since there is a y factor, one of our multiplicands needs to be either y or P_{yx}. Choosing y will not help because we do not have an x^{n+1} to make f, so one of our multiplicands is definitely P_{yx}. The only choice for a second multiplicand that can give us f is one from P_x. Multiplying these terms gives us terms of the form $zyx^i + \mathrm{c}yx^{i+1} + y\sum_{j\in S} x^{n+1-j+i}$, which includes yx^{n+1} if $i \in S$, but then we have to be able to produce the term zyx^i for some $i \in S$ to be able to cancel it out.

To get zyx^i, using only two multiplicands, we have the following four possibilities:

- use y and zx^i to get yzx^i for some $i \in C$, but since $C \cap S = \varnothing$ we can not get yzx^i for any $i \in S$.
- use x^i and P_{yz} again, but this cancels out our desired term yx^{n+1} as well since we have to use the same i.
- use z and P_{yx} to get $z^2y + \mathrm{c}xyz + zy\sum_{j\in S} x^{n+1-j}$, which includes zyx^i if $n+1-i \in S$ or if $i = 1$, but then, in either case, we have to cancel z^2y and the only way to get z^2y is to use the same terms again which cancels our desired term zyx^i as well.
- use P_{zx} and P_{yx} to get $z^2x^ky + \mathrm{c}x^{k+1}yz + zy\sum_{j\in S} x^{n+1-j+k}$, which includes zyx^i if $n+1-i+k \in S$ or if $k+1 = i$, but then, in either case, we have to cancel z^2x^ky and the only way to get z^2x^ky is to use the same terms again with the same k which cancels our desired term zyx^i as well.

Hence f is independent of $(P \parallel P', Q)$ and the proof of Claim D is complete. □

Stage 2: Now that we have proved our specific assumption is an OBDHE assumption, we prove that under this assumption, the GKEA assumption, and the UOWHF assumption OurBE is adaptive CCA secure.

Let A be an adaptive CCA adversary for OurBE. We construct an adversary B that successfully breaks our specific assumption, if A is successful in its attack against OurBE, the GKEA assumption holds, and H is a UOWHF.

First of all, note that, based on Lemma 6 and a discussion similar to that of Stage 1, as long as H_κ is a UOWHF, it can be indistinguishably simulated *independently* of its input in the generic group model, and hence hashed values can be considered *constant* for this proof. Jumping ahead, we treat $\mathrm{c} = \mathsf{H}_\kappa(C_0)$ and $\mathrm{c}^* = \mathsf{H}_\kappa(C_0^*)$ as constant coefficients for polynomials.

Let B be given the following quantities:

$$g,\ h,\ \{g_k = g^{\alpha^k}\}_{k\in\{1,\ldots,2n\}\setminus\{n+1\}},\ v,\ T,$$

and (restricted) oracle access to $\mathcal{O}^{\mathrm{DH}}_{g,e}$ as specified by the assumption. It is supposed to distinguish whether $T = e(g^{\alpha^{n+1}}, h)$ or T is random. As a UOWHF adversary, B gives out h as the first message on which it wishes to be challenged and receives a key κ for the hash function. B runs A on input $EK = (g, v, \kappa)$.

On a join query for user i made by the adversary, B gives $pk_i = (g_i, g_{n+1-i}, g_{n+1+i})$ to A.

On any private key query for user i made by A, B queries the oracle $\mathcal{O}^{\mathrm{DH}}_{g,e}(g_i, v)$ and and gives the oracle output to A. Note that if we assume $v = g^\gamma$, then the oracle output is equal to g_i^γ.

On a decryption oracle query (i, S, H), where $H = (C_0, C_1)$, B first checks the ciphertext validity. If the ciphertext is invalid it replies with $\perp$. Let $\mathrm{c} = \mathsf{H}_\kappa(C_0)$. If the ciphertext is valid, then it is in the following form:

$$H = (C_0, C_0^q), \qquad \text{where} \quad q = \gamma + \mathrm{c}\alpha + \sum_{j\in S} \alpha^{n+1-j}. \tag{3}$$

Let us assume, without loss of generality, that all the potential $n-1$ users are initiated. Let C denote the set of corrupted users by A and $N^* = \{1, \ldots, 2n\} \setminus \{n+1\}$. Now A can be viewed as an algorithm that on input g, v, κ, $\{g_i\}_{i\in N^*}$, and $\{d_i\}_{i\in C}$ outputs $H = (C_0, C_0^q)$ as above. Note that the input to A (excluding $\kappa \notin \mathbb{G}$) can be written as follows:

$$g^P, \qquad \text{where} \quad P = (\ 1,\ \gamma,\ \{\alpha^i\}_{i\in N^*},\ \{\gamma\alpha^i\}_{i\in C}\ ).$$

To apply the GKEA assumption, note that here $\mathrm{Span}(P)$ includes all the elements of the following form:

$$\rho = u + x\gamma + \sum_{i\in N^*} y_i\alpha^i + \gamma\sum_{i\in C} z_i\alpha^i, \qquad \text{for random } u, x, y_i, z_i. \tag{4}$$

Consider ρq for some ρ and the q defined above, respectively in Equations 3 and 4. For ρq to be in $\mathrm{Span}(P)$, we should have $x = 0$ and $\forall i \in C : z_i = 0$ because otherwise ρq will have either the factor

$x\gamma^2$ or $z_i\gamma^2\alpha^i$ for some i and would not fall in Span(P). Hence any ρ satisfying $\rho q \in$ Span(P) should be in the form

$$\rho = u + \sum_{i \in N^*} y_i \alpha^i, \qquad \text{for random } u, y_i. \tag{5}$$

A basis for such a subspace is the set $\{1, \{\alpha^i\}_{i\in N^*}\}$. Therefore the GKEA assumption guarantees that there exists an extractor that outputs the values $\{\beta, \{b_i\}_{i\in N^*}\}$ such that

$$C_0 = g^{\beta + \sum_{i \in N^*} b_i \alpha^i} = g^\beta \prod_{i \in N^*} g_i^{b_i}.$$

Now note that $K = e(g_{n+1}, C_0)$. Hence the session key can be calculated based on the known representation of C_0 in terms of g and g_i, e.g., as follows:

$$\begin{aligned} K &= e(g_{n+1}, g^\beta \prod_{i \in N^*} g_i^{b_i}) = e(g_{n+1}, g)^\beta \prod_{i \in N^*} e(g_{n+1}, g_i)^{b_i} \\ &= e(g_n, g_1)^\beta e(g_{2n}, g_1)^{b_n} e(g_{n+2}, g_{2n-1})^{b_{2n}} \prod_{i \in N^* \setminus \{n, 2n\}} e(g_n, g_{i+1})^{b_i}. \end{aligned}$$

At some point, the adversary A terminates the first query phase and outputs a set S^* on which it wants to be challenged. B calculates $w = v g_1^{\mathsf{H}_\kappa(h)} \prod_{j \in S^*} g_{n+1-j}$, makes the oracle query $\mathcal{O}^{\mathrm{DH}}_{g,e}(w, h)$, receives the oracle output h', sets the challenge ciphertext as $H^* = (C_0^*, C_1^*) = (h, h')$, and gives H^* along with $K = T$ to A. Let $\mathrm{c}^* = \mathsf{H}_\kappa(C_0^*)$. Note that, Equation 1 (see page 5) holds, hence C is a valid ciphertext, and C_1^* should be equal to C_0^* raised to a power of the following form:

$$\gamma + \mathrm{c}^*\alpha + \sum_{j \in S^*} \alpha^{n+1-j}.$$

Furthermore, if $T = e(g_{n+1}, h)$, then K is the correct key corresponding to the ciphertext H^*, and if T is random, then K is a random key.

In the second phase of the attack, B answers A's join and corruption oracle queries as in the first phase, and A's decryption oracle queries, in a fashion similar to that of prior to the challenge, as follows.

On a decryption oracle query (i, S, H), where $H = (C_0, C_1)$, B first checks its validity, and if valid, it is in the form of Equation 3.

Now the input to A can be listed as g, v, κ, $\{g_i\}_{i\in N^*}$, and $\{d_i\}_{i\in C}$, plus $H^* = (C_0^*, C_1^*)$. Let $C_0^* = g^{t^*}$. The input to A can be written as g^P, where P is as follows (κ, K_0, and K_1 can be disregarded as they are not in $\mathbb{G}$):

$$P = (\ 1,\ \gamma,\ \{\alpha^i\}_{i\in N^*},\ \{\gamma\alpha^i\}_{i\in C},\ t^*,\ t^*(\gamma + \mathrm{c}^*\alpha + \sum_{j\in S^*} \alpha^{n+1-j})\ ).$$

Span(P) includes all the linear combinations ρ of the above terms. Similarly, we argue that ρ cannot include any γ or $\gamma\alpha^i$ terms because they would induce γ^2 or $\gamma^2\alpha^i$ terms respectively in the product ρq. Furthermore, ρ cannot include the last term because it would induce a non-cancelable $t^*\gamma^2$ term in the product ρq. In addition, note that if ρ includes the term t^*, then ρq would include the term

$$t^*(\gamma + \mathrm{c}\alpha + \sum_{j\in S} \alpha^{n+1-j}).$$

The only way a ρ including this term can be contained in Span(P) is if $\mathrm{c}^* = \mathrm{c}$ (i.e., $\mathsf{H}_\kappa(C_0^*) = \mathsf{H}_\kappa(C_0)$) and $S = S^*$ (note that $j \le n-1$, so $n+1-j \ge 2$), which contradicts H being a UOWHF. Therefore, ρ cannot include the term t^*, and again ρ should be in the form of Equation 5, and hence the session key can be calculated similarly as before.

At the end, A outputs its guess b. B outputs b as its decision for its received challenge. Based on the above discussion, if A is successful in its adaptive CCA attack, then B is able to either contradict H being a UOWHF or distinguish $T = e(g_{n+1}, h)$ from a random element successfully. Hence the proof of Theorem 4 is complete. □