An extended abstract of this paper appears in CRYPTO '03 — Dan Boneh (editor), Proceedings of the 23rd International Conference on Advances in Cryptology, Lecture Notes in Computer Science, volume 2729, pages 383–399, Berlin: Springer-Verlag, 2003. This version is available at: `http://eprint.iacr.org/2003/163`.

Multipurpose Identity-Based Signcryption
A Swiss Army Knife for Identity-Based Cryptography

Xavier Boyen

Abstract

Identity-Based (IB) cryptography is a rapidly emerging approach to public-key cryptography that does not require principals to pre-compute key pairs and obtain certificates for their public keys—instead, public keys can be arbitrary identifiers such as email addresses, while private keys are derived at any time by a trusted private key generator upon request by the designated principals. Despite the flurry of recent results on IB encryption and signature, some questions regarding the security and efficiency of practicing IB encryption (IBE) and signature (IBS) as a joint IB signature/encryption (IBSE) scheme with a common set of parameters and keys, remain unanswered.

We first propose a stringent security model for IBSE schemes. We require the usual strong security properties of: (for confidentiality) indistinguishability against adaptive chosen-ciphertext attacks, and (for non-repudiation) existential unforgeability against chosen-message insider attacks. In addition, to ensure as strong as possible ciphertext armoring, we also ask (for anonymity) that authorship not be transmitted in the clear, and (for unlinkability) that it remain unverifiable by anyone except (for authentication) by the legitimate recipient alone.

We then present an efficient IBSE construction, based on bilinear pairings, that satisfies all these security requirements, and yet is as compact as pairing-based IBE and IBS in isolation. Our scheme is secure, compact, fast and practical, offers detachable signatures, and supports multi-recipient encryption with signature sharing for maximum scalability.

1 Introduction

Recently, Boneh and Franklin [5] observed that bilinear pairings on elliptic curves could be used to make identity-based encryption possible and practical. Following this seminal insight, the last couple of years have seen a flurry of results on a number of aspects of what has now become the nascent field of Identity-Based (IB) cryptography.

1.1 Identity-Based Cryptography

The distinguishing characteristic of IB cryptography is the ability to use any string as a public key; the corresponding private key can only be derived by a trusted Private Key Generator (PKG), custodian of a master secret. For encryption purposes, this allows Alice to securely send Bob an encrypted message, using as public key any unambiguous name identifying Bob, such as Bob's email address, possibly before Bob even knows his own private key. For signature purposes, Alice may sign her communications using a private key that corresponds to an unambiguous name of hers, so that anybody can verify the authenticity of the signature simply from the name, without the need for a certificate. Revocation issues are handled by using short-lived time-dependent identities [5].

[0]Currently with Voltage Security, Palo Alto, California — *xb@boyen.org*

An inherent limitation of IB cryptography is the trust requirement that is placed on the PKG, as an untrustworthy PKG will have the power to forge Alice's signature, and decrypt Bob's past and future private communications. This can be partially alleviated by splitting the master secret among several PKGs under the jurisdiction of no single entity, as explained in [5]. The window of vulnerability can also be reduced by periodically changing the public parameters, and purging any master secret beyond a certain age, effectively limiting the interval during which IB cryptograms can be decrypted. Traditional public-key cryptography is not completely immune to the problem, either: in a public key infrastructure, the certification authority has the power to issue fake certificates and impersonate any user for signature purposes; it can similarly spoof encryption public key certificates in order to decrypt future ciphertexts addressed to targeted users, albeit not in a manner not amenable to easy detection.

The idea of IB cryptography first emerged in 1984 [24], although only an IB signature (IBS) scheme was then suggested, based on conventional algebraic methods in $\mathbb{Z}_n$. Other IBS and identification schemes were quick to follow [13, 12]. However, it is only in 2001 that a practical IB encryption (IBE) mechanism was finally suggested [5], based on the much heavier machinery of bilinear pairings on elliptic curves, whose use in cryptography had slowly started to surface in the few years prior, *e.g.*, for key exchange [18] and IBS [23]. Interestingly, a more conventional approach to IBE was proposed shortly thereafter [10], albeit not as efficient as the pairing-based IBE scheme of Boneh and Franklin.

1.2 Motivation and Contribution

Following the original publication of the BF-IBE scheme [5], a number of authors have proposed various new applications of pairing-based IB cryptography. These include various IB signature schemes [22, 16, 8], key agreement [25], a 2-level hierarchical IB encryption [17], and a general hierarchical IB encryption that can also be used to produce signatures [15]. More specifically focusing on joint authentication and encryption, we note a repudiable authenticated IBE [20], an authenticated key agreement scheme [9], and a couple of IB signcryption schemes that efficiently combine signature and encryption [21, 19].

What the picture is currently missing is an algorithm that combines (existing or new) IBE and IBS in a practical and secure way. Indeed, it would be of great practical interest to be able to use the same IB infrastructure for signing and encrypting. A possibility is to combine some existing IBE and IBS using black-box composition techniques, such as [1]; this is however rather suboptimal.

A better approach would be to exploit the similarities between IBE and IBS, and elaborate a dual-purpose IB Encryption-Signature (IBSE) scheme based on a shared infrastructure, toward efficiency increases and security improvements. Doing so, we would have to ensure that no hidden weakness arises from the combination, which is always a risk if the same parameters and keys are used. The issues that arise from this approach are summarized as follows:

- Can IBE and IBS be practiced in conjunction, in a secure manner, sharing infrastructure, parameters, and keys, toward greater efficiency?
- What emerging security properties can be gained from such a combination?

Our contributions to answering these questions are twofold. We first specify a security model that a strong IBSE combination should satisfy. Our model specifies the IBSE version of the strongest notions of security usually considered in public-key cryptography. For confidentiality, we define a notion of ciphertext indistinguishability under adaptive chosen-ciphertext attacks. For non-repudiation, we define a notion of signature unforgeability under chosen-message attacks, in the stringent case of an 'insider' adversary, *i.e.*, with access to the decryption private key, as considered in [1]. We also specify the additional security features of ciphertext authentication, anonymity, and unlinkability, that, if less conventional, are highly desirable in practice: together, they convince the legitimate recipient of the ciphertext origin, and conceal it from anyone else.

We then propose a fast IBSE scheme satisfying our strong security requirements, which we prove in the random oracle model [3]. Our scheme uses the properties of bilinear pairings to achieve a two-layer sign-then-encrypt combination, featuring a detachable randomized signature, followed by anonymous deterministic encryption. The scheme is very efficient, more secure than what we call *monolithic* signcryption—in which a single operation is used for decryption and signature verification, as in the original signcryption model of [27]—and more compact than generic compositions of IBE and IBS. Our two-layer design is also readily adapted to provide multi-recipient encryption of the same message with a shared signature and a single bulk message encryption.

Performance-wise, our dual-purpose optimized IBSE scheme is as compact as most existing single-purpose IBE and IBS taken in isolation. It is also about as efficient as the monolithic IB signcryption schemes of [21] and [19], with the added flexibility and security benefits that separate anonymous decryption and signature verification layers can provide. A comparative summary of our scheme with competing approaches can be found in §6, Table 2.

1.3 Outline of the Paper

We start in §2 by laying out the abstract IBSE specifications. In §3, we formalize the various security properties sought from the cryptosystem. In §4, we review the principles of IB cryptography based on pairings. In §5, we describe an implementation of our scheme. In §6, we make detailed performance and security comparisons with the competition. In §7, we prove compliance of our implementation with the security model. In §8, we study a few extensions of practical significance. Finally, in §9, we draw some conclusions.

2 Specification of the Cryptosystem

An *Identity-Based Signature/Encryption* scheme, or IBSE, consists of a suite of six algorithms: Setup, Extract, Sign, Encrypt, Decrypt, and Verify. In essence, Setup generates random instances of the common public parameters and master secret; Extract computes the private key corresponding to a given public identity string; Sign produces a signature for a given message and private key; Encrypt encrypts a signed plaintext for a given identity; Decrypt decrypts a ciphertext using a given private key; Verify checks the validity of a given signature for a given message and identity. Messages are arbitrary strings in $\{0,1\}^*$.

The functions that compose a generic IBSE are thus specified as follows.

Setup On input 1^n, produces a pair $\langle \sigma, \pi \rangle$ (where σ is a randomly generated master secret and π the corresponding common public parameters, for the security meta-parameter n).

$\mathsf{Extract}_{\pi,\sigma}$ On input id, computes a private key pvk (corresponding to the identity id under $\langle \sigma, \pi \rangle$).

Sign_{π} On input $\langle \mathsf{pvk}_A, \mathsf{id}_A, m \rangle$, outputs a signature s (for pvk_A, under π), and some ephemeral state data r.

$\mathsf{Encrypt}_{\pi}$ On input $\langle \mathsf{pvk}_A, \mathsf{id}_B, m, s, r \rangle$, outputs an anonymous ciphertext c (containing the signed message $\langle m, s \rangle$, encrypted for the identity id_B under π).

$\mathsf{Decrypt}_{\pi}$ On input $\langle \mathsf{pvk}_B, \hat{c} \rangle$, outputs a triple $\langle \hat{\mathsf{id}}_A, \hat{m}, \hat{s} \rangle$ (containing the purported sender identity and signed message obtained by decrypting $\hat{c}$ by the private key pvk_B under π).

Verify_π On input $\langle \hat{\mathsf{id}}_A, \hat{m}, \hat{s} \rangle$, outputs $\top$ 'true' or $\bot$ 'false' (indicating whether $\hat{s}$ is a valid signature for the message $\hat{m}$ by the identity $\hat{\mathsf{id}}_A$, under π).

Since we are concerned with sending messages that are simultaneously encrypted and signed, we allow the encryption function to make use of the private key of the sender. Accordingly, we assume that Encrypt is always used on an output from Sign, so that we may view the Sign/Encrypt composition as a single 'signcryption' function; we keep them separate to facilitate the treatment of multi-recipient encryption with shared signature in §8.1. We also insist on the dichotomy Decrypt *vs.* Verify, to permit the decryption of anonymous ciphertexts, and to decouple signature verification from the data that is transmitted over the wire, neither of which would be feasible had we used a monolithic 'unsigncryption' function.

It is required that these algorithms jointly satisfy the following consistency constraints.

Definition 1. For all master secret and common parameters $\langle \sigma, \pi \rangle \leftarrow \mathsf{Setup}[1^n]$, any identities id_A and id_B, and matching private keys $\mathsf{pvk}_A = \mathsf{Extract}_{\pi,\sigma}[\mathsf{id}_A]$ and $\mathsf{pvk}_B = \mathsf{Extract}_{\pi,\sigma}[\mathsf{id}_B]$, we require for consistency that, $\forall m \in \{0,1\}^*$:

$$\left\{ \begin{array}{rcl} \langle s, r \rangle & \leftarrow & \mathsf{Sign}_\pi[\mathsf{pvk}_A, \mathsf{id}_A, m] \\ c & \leftarrow & \mathsf{Encrypt}_\pi[\mathsf{pvk}_A, \mathsf{id}_B, m, s, r] \\ \langle \hat{\mathsf{id}}_A, \hat{m}, \hat{s} \rangle & \leftarrow & \mathsf{Decrypt}_\pi[\mathsf{pvk}_B, c] \end{array} \right\} \Longrightarrow \left\{ \begin{array}{rcl} \hat{\mathsf{id}}_A & = & \mathsf{id}_A \\ \hat{m} & = & m \\ \mathsf{Verify}_\pi[\mathsf{id}_A, \hat{m}, \hat{s}] & = & \top \end{array} \right\}$$

In the sequel, we omit the subscripted parameters π and σ when understood from context.

3 Formal Security Model

Due to the identity-based nature of our scheme, and the combined requirements on confidentiality and non-repudiation, the security requirements are multifaceted and quite stringent. For example, for confidentiality purposes, one should assume that the adversary may obtain any private key other than that of the targeted recipient, and has an oracle that decrypts any valid ciphertext other than the challenge. For non-repudiation purposes, we assume that the forger has access to any private key other than that of the signer, and can query an oracle that signs and encrypts any message but the challenge. These assumptions essentially amount to the 'insider' model in the terminology of [1].

We also consider the notions of *ciphertext unlinkability* and *ciphertext authentication*, which allow the legitimate recipient to privately verify—but not prove to others—that the ciphertext addressed to him and the signed message it contains were indeed produced by the same entity. We note that these properties are not jointly achieved by other schemes that combine confidentiality and non-repudiation, such as the signcryption of [21] and [19]. We also ask for *ciphertext anonymity*, which simply means that no third party should be able to discover whom a ciphertext originates from or is addressed to, if the sender and recipient wish to keep that a secret.

All these properties are recapitulated as follows.

1. *message confidentiality* (§3.1): allows the communicating parties to preserve the secrecy of their exchange, if they choose to.
2. *signature non-repudiation* (§3.2): makes it universally verifiable that a message speaks in the name of the signer (regardless of the ciphertext used to convey it, if any). This implies message authentication and integrity.
3. *ciphertext unlinkability* (§3.3): allows the sender to disavow creating a ciphertext for any given recipient, even though he or she remains bound to the valid signed message it contains.

4. *ciphertext authentication* (§3.4): allows the legitimate recipient, alone, to be convinced that the ciphertext and the signed message it contains were crafted by the same entity. This implies ciphertext integrity. It also reassures the recipient that the communication was indeed secured end-to-end.
5. *ciphertext anonymity* (§3.5): makes the ciphertext appear anonymous (hiding both the sender and the recipient identities) to anyone who does not possess the recipient decryption key.

For simplicity of the subsequent analysis, we disallow messages from being addressed to the same identity as authored them—a requirement that we call the *irreflexivity assumption*. Remark that if such a mode of operation is nonetheless desired, it can easily be achieved, either, (1) by endowing each person with an additional 'self' identity, under which they can encrypt messages signed under their regular identity, or, (2) by splitting each identity into a 'sender' identity and a 'recipient' identity, to be respectively used for signature and encryption purposes. This can be done, *e.g.*, by prepending an indicator bit to all identity strings; each individual would then be given two private keys by the PKG.

For clarity, and regardless of which of the above convention is chosen, if any, we use the subscripts 'A' for Alice the sender and 'B' for Bob the recipient.

3.1 Message Confidentiality

Message confidentiality against adaptive chosen-ciphertext attacks is defined in terms of the following game, played between a challenger and an adversary. We combine signature and encryption into a dual-purpose oracle, to allow $\mathsf{Encrypt}$ to access the ephemeral random state data r from Sign.

Start The challenger runs the Setup procedure for a given value of the security parameter n, and provides the common public parameters π to the adversary, keeping the secret σ for itself.

Phase 1 The adversary makes a number of queries to the challenger, in an adaptive fashion (*i.e.*, one at a time, with knowledge of the previous replies). The following queries are allowed:

signature/encryption queries in which the adversary submits a message and two distinct identities, and obtains a ciphertext containing the message signed in the name of the first identity and encrypted for the second identity;

decryption queries in which the adversary submits a ciphertext and an identity, and obtains the identity of the sender, the decrypted message, and a valid signature, provided that (1) the decrypted identity of the sender differs from that of the specified recipient, and (2) the signature verification condition $\mathsf{Verify} = \top$ is satisfied; otherwise, the oracle only indicates that the ciphertext is invalid for the specified recipient;

private key extraction queries in which the adversary submits an identity, and obtains the corresponding private key;

Selection At some point, the adversary returns two distinct messages m_0 and m_1 (assumed of equal length), a signer identity id_A, and a recipient identity id_B, on which it wishes to be challenged. The adversary must have made no private key extraction query on id_B.

Challenge The challenger flips $b \in \{0, 1\}$, computes $\mathsf{pvk}_A = \mathsf{Extract}[\mathsf{id}_A]$, $\langle s, r\rangle \leftarrow \mathsf{Sign}[\mathsf{pvk}_A, m_b]$, $c \leftarrow \mathsf{Encrypt}[\mathsf{pvk}_A, \mathsf{id}_B, m_b, s, r]$, and returns the ciphertext c as challenge to the adversary.

Phase 2 The adversary adaptatively issues a number of additional encryption, decryption, and extraction queries, under the additional constraint that it not ask for the private key of id_B or the decryption of c under id_B.

Response The adversary returns a guess $\hat{b} \in \{0,1\}$, and wins the game if $\hat{b} = b$.

It is emphasized that the adversary is allowed to know the private key pvk_A corresponding to the signing identity, which gives us *insider-security* for confidentiality [1]. On the one hand, this is necessary if confidentiality is to be preserved in case the sender's private key becomes compromised. On the other hand, this will come handy when we study a 'repudiable' IBSE variant in §8.2.

This game is very similar to the IND-ID-CCA attack in [5]; we call it an IND-IBSE-CCA attack.

Definition 2. An identity-based joint encryption and signature (IBSE) scheme is said to be semantically secure against adaptive chosen-ciphertext insider attacks, or *IND-IBSE-CCA secure*, if no randomized polynomial-time adversary has a non-negligible advantage in the above game. In other words, any randomized polynomial-time IND-IBSE-CCA adversary $\mathcal{A}$ has an advantage $\mathbf{Adv}_{\mathcal{A}}[n] = |\mathbf{P}[\hat{b} = b] - \frac{1}{2}|$ that is $o[1/\mathrm{poly}[n]]$ for any polynomial $\mathrm{poly}[n]$ in the security parameter.

Remark that we insist that the decryption oracle perform a validity check before returning a decryption result, even though $\mathsf{Decrypt}$ does not specify it. This requirement hardly weakens the model, and allows for stronger security results. We similarly ask that the oracles enforce the irreflexivity assumption, *e.g.*, by refusing to produce or decrypt non-compliant ciphertexts.

3.2 Signature Non-Repudiation

Signature non-repudiation is formally defined in terms of the following game, played between a challenger and an adversary.

Start The challenger runs the Setup procedure for a given value of the security parameter n, and provides the common public parameters π to the adversary, keeping the secret σ for itself.

Query The adversary makes a number of queries to the challenger. The attack may be conducted adaptively, and allows the same queries as in the Confidentiality game of §3.1, namely: signature/encryption queries, decryption queries, and private key extraction queries.

Forgery The adversary returns a recipient identity id_B and a ciphertext c.

Outcome The adversary wins the game if the ciphertext c decrypts, under the private key of id_B, to a signed message $\langle \mathsf{id}_A, \hat{m}, \hat{s} \rangle$ that satisfies $\mathsf{id}_A \neq \mathsf{id}_B$ and $\mathsf{Verify}[\mathsf{id}_A, \hat{m}, \hat{s}] = \top$, provided that (1) no private key extraction query was made on id_A, and (2) no signature/encryption query was made that involved $\hat{m}$, id_A, and some recipient $\mathsf{id}_{B'}$, and resulted in a ciphertext c' whose decryption under the private key of $\mathsf{id}_{B'}$ is the claimed forgery $\langle \mathsf{id}_A, \hat{m}, \hat{s} \rangle$.

Such a model is very similar to the usual notion of existential unforgeability against chosen-message attacks [11, 26]; we call it an EUF-IBSE-CMA attack.

Definition 3. An IBSE scheme is said to be existentially signature-unforgeable against chosen-message insider attacks, or *EUF-IBSE-CMA secure*, if no randomized polynomial-time adversary has a non-negligible advantage in the above game. In other words, any randomized polynomial-time EUF-IBSE-CMA adversary $\mathcal{A}$ has an advantage $\mathbf{Adv}_{\mathcal{A}}[n] = \mathbf{P}[\mathsf{Verify}[\mathsf{id}_A, \hat{m}, \hat{s}] = \top]$ that behaves as $o[1/\mathrm{poly}[n]]$ for any polynomial $\mathrm{poly}[n]$.

In the above experiment, the adversary is allowed to obtain the private key pvk_B for the forged message recipient id_B, which corresponds to the stringent requirements of *insider-security* for authentication [1]. There is one important difference, however: in [1], non-repudiation applies to the ciphertext itself, which is the only sensible thing to do in the context of a signcryption model with a monolithic 'unsigncryption' function. Here, given our two-step $\mathsf{Decrypt}/\mathsf{Verify}$ specification, we define non-repudiation with respect to the decrypted signature, which is more intuitive and does not preclude ciphertext unlinkability (see §3.3).

3.3 Ciphertext Unlinkability

Ciphertext unlinkability is the property that makes it possible for Alice to deny having sent a given ciphertext to Bob, even if the ciphertext decrypts (under Bob's private key) to a message bearing Alice's signature. In other words, the signature should only be a proof of authorship of the plaintext message, and not the ciphertext. (We shall make one exception to this requirement in §3.4, where we seek that the legitimate recipient be able privately authenticate the ciphertext, in order to be convinced that it is indeed addressed to him or her.)

Ciphertext unlinkability allows Alice, *e.g.*, as a news correspondent in a hostile area, to stand behind the content of her reporting, but conceal any detail regarding the particular channel, method, place, or time of communication, lest subsequent forensic investigations be damaging to her sources. When used in conjunction with the multi-recipient technique of §8.1, this property also allows her to deniably provide exact copies of her writings to additional recipients.

We do not present a formal experiment for this property. Suffice it to say that it is enough to ask that, given a plaintext message signed by Alice, Bob be able to create a valid ciphertext addressed to himself for that message, that is indistinguishable from a genuine ciphertext from Alice.

Definition 4. An IBSE scheme is said to be ciphertext-unlinkable if there exists a polynomial-time algorithm that, given an identified signed message $\langle \mathsf{id}_A, m, s\rangle$ such that $\mathsf{Verify}[\mathsf{id}_A, m, s] = \top$, and a private key $d_B = \mathsf{Extract}[\mathsf{id}_B]$, assembles a ciphertext c that is computationally indistinguishable from a genuine encryption of $\langle m, s\rangle$ by id_A for id_B.

As mentioned earlier, ciphertext unlinkability is the reason why we considered the notion of signature unforgeability in §3.2, instead of the usual notion of ciphertext unforgeability as studied in the signcryption model of [1]. Indeed, if a ciphertext were unforgeable, surely it would be undeniably linkable to its author.

Note also that ciphertext unlinkability only makes sense in a two-layer signcryption model like ours, as opposed to the monolithic model of [27] used in [21, 19]. Indeed, if part of the ciphertext itself is needed to verify the authenticity of the plaintext, ciphertext indistinguishability is lost as soon as the recipient is compelled to prove authenticity to a third party.

3.4 Ciphertext Authentication

Ciphertext authentication is, in a sense, the complement to unlinkability. Authentication requires that the legitimate recipient be able to ascertain that the ciphertext did indeed come from the same person who signed the message it contains. (Naturally, he or she cannot prove this to anyone else, per the unlinkability property.)

Another use of ciphertext authentication is to convince the recipient that the ciphertext was encrypted throughout the entire transmission. In particular, a ciphertext properly authenticated in this model cannot have been the target of a (successful, active) man-in-the-middle interception.

We define ciphertext authentication in terms of the following game.

Start The challenger runs the Setup procedure for a given value of the security parameter n, and provides the common public parameters π to the adversary, keeping the secret σ for itself.

Query The adversary makes a number of queries to the challenger, as in the Confidentiality game of §3.1 and the Non-repudiation game of §3.2.

Forgery The adversary returns a recipient identity id_B and and a ciphertext c.

Outcome The adversary wins the game if c decrypts, under the private key of id_B, to a signed message $\langle \mathsf{id}_A, \hat{m}, \hat{s} \rangle$ such that $\mathsf{id}_A \neq \mathsf{id}_B$ and that satisfies $\mathsf{Verify}[\mathsf{id}_A, \hat{m}, \hat{s}] = \top$, provided that (1) no private key extraction query was made on either id_A or id_B, and (2) c did not result from a signature/encryption query with sender and recipient identities id_A and id_B.

We contrast the above experiment, which is a case of 'outsider' security for authentication on the whole ciphertext, with the scenario for signature non-repudiation, which required insider security on the signed plaintext only. We call the above experiment an AUTH-IBSE-CMA attack.

Definition 5. An IBSE scheme is said to be existentially ciphertext-unforgeable against chosen-message outsider attacks, or *AUTH-IBSE-CMA secure*, if no randomized polynomial-time adversary has a non-negligible advantage in the above game. In other words, any randomized polynomial-time EUF-IBSE-CMA adversary $\mathcal{A}$ has an advantage $\mathbf{Adv}_{\mathcal{A}}[n] = \mathbf{P}[\mathsf{Verify}[\mathsf{id}_A, \hat{m}, \hat{s}] = \top]$ that behaves as $o[1/\mathrm{poly}[n]]$ for any polynomial $\mathrm{poly}[n]$.

3.5 Ciphertext Anonymity

Finally, we require ciphertext anonymity, which is to say that the ciphertext must contain no information in the clear that identifies the author or recipient of the message (and yet be decipherable by the intended recipient without that information).

Ciphertext anonymity against adaptive chosen-ciphertext attacks is defined as follows.

Start The challenger runs the Setup procedure for a given value of the security parameter n, and provides the common public parameters π to the adversary, keeping the secret σ for itself.

Phase 1 The adversary is allowed to make adaptive queries of the same types as in the Confidentiality game of §3.1, *i.e.*: signature/encryption queries, decryption queries, and private key extraction queries.

Selection At some point, the adversary returns a message m, two sender identities id_{A_0} and id_{A_1}, and two recipient identities id_{B_0} and id_{B_1}, on which it wishes to be challenged. The adversary must have made no private key extraction query on either id_{B_0} or id_{B_1}.

Challenge The challenger flips two random coins $b', b'' \in \{0, 1\}$, computes $\mathsf{pvk} = \mathsf{Extract}[\mathsf{id}_{A_{b'}}]$, $\langle s, r \rangle \leftarrow \mathsf{Sign}[\mathsf{pvk}, m]$, $c \leftarrow \mathsf{Encrypt}[\mathsf{pvk}, \mathsf{id}_{B_{b''}}, m, s, r]$, and gives the ciphertext c to the adversary.

Phase 2 The adversary adaptatively issues a number of additional encryption, decryption, and extraction queries, under the additional constraint that it not ask for the private key of either id_{B_0} or id_{B_1}, or the decryption of c under id_{B_0} or id_{B_1}.

Response The adversary returns two guesses $\hat{b}', \hat{b}'' \in \{0, 1\}$, and wins the game if $\langle \hat{b}', \hat{b}'' \rangle = \langle b', b'' \rangle$.

This game is the same as for confidentiality, except that the adversary is challenged on the identities instead of the message; it is an insider attack. We call it an ANON-IBSE-CCA attack.

Definition 6. An IBSE is said to be ciphertext-anonymous against adaptive chosen-ciphertext insider attacks, or *ANON-IBSE-CCA secure*, if no randomized polynomial-time adversary has a non-negligible advantage in the above game. In other words, any randomized polynomial-time ANON-IBSE-CCA adversary $\mathcal{A}$ has an advantage $\mathbf{Adv}_{\mathcal{A}}[n] = |\mathbf{P}[\hat{b} = b] - \frac{1}{4}|$ that is $o[1/\mathrm{poly}[n]]$ for any polynomial $\mathrm{poly}[n]$ in the security parameter.

We emphasize that anonymity only applies to the ciphertext, against non-recipients, and is thus consistent with both non-repudiation (§3.2) and authentication (§3.4). To illustrate the difference bewteen unlinkability and anonymity, note that the authenticated IBE scheme of [20] is unlinkable but not anonymous, since the sender identity must be known prior to decryption.

Ciphertext anonymity and ciphertext unlinkability are two properties that are unattainable in the monolithic signcryption model.

4 Review of IB Cryptography from Pairings

We now give a brief summary of the Boneh-Franklin algorithm for identity-based cryptography based on bilinear pairings on elliptic curves.

Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two cyclic groups of prime order p, writing the group action multiplicatively (in both cases using 1 to denote the neutral element).

Definition 7. An (efficiently computable, non-degenerate) map $\mathbf{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ is called a *bilinear pairing* if, for all $x, y \in \mathbb{G}_1$ and all $a, b \in \mathbb{Z}$, we have $\mathbf{e}[x^a, y^b] = \mathbf{e}[x, y]^{a\,b}$.

Definition 8. The (computational) *bilinear Diffie-Hellman problem* for a bilinear pairing as above is described as follows: given $g, g^a, g^b, g^c \in \mathbb{G}_1$, where g is a generator and $a, b, c \in \mathbb{F}_p^\star$ are chosen at random, compute $\mathbf{e}[g, g]^{a\,b\,c}$. The advantage of an algorithm $\mathcal{B}$ at solving the BDH problem is defined as $\mathbf{Adv}_{\mathcal{B}}[\mathbf{e}] = \mathbf{P}[\mathcal{B}[g, g^a, g^b, g^c] = \mathbf{e}[g, g]^{a\,b\,c}]$.

Definition 9. Let $\mathcal{G}$ be a polynomial-time randomized function that, on input 1^n, returns the description of a bilinear pairing $\mathbf{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ between two groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of prime order p. A BDH parameter generator $\mathcal{G}$ satisfies the *bilinear Diffie-Hellman assumption* if there is no randomized algorithm $\mathcal{B}$ that solves the BDH problem in time $\mathcal{O}[\mathrm{poly}[n]]$ with advantage $\Omega[1/\mathrm{poly}[n]]$. The probability space is that of the randomly generated parameters $\langle \mathbb{G}_1, \mathbb{G}_2, p, \mathbf{e}\rangle$, the BDH instances $\langle g, g^a, g^b, g^c\rangle$, and the randomized executions of $\mathcal{B}$.

The Boneh-Franklin system provides a concrete realization of the above definitions. It is based on an elliptic-curve implementation of the BDH parameter generator $\mathcal{G}$, which we describe following [2, 14] as recently generalized in [6].

Let $E/\mathbb{F}_q$ be an elliptic curve defined over some ground field $\mathbb{F}_q$ of prime characteristic χ. For any extension degree $r \geq 1$, let $E(\mathbb{F}_{q^r})$ be the group of points in $\{\langle x, y\rangle \in (\mathbb{F}_{q^r})^2\} \cup \{\infty\}$ that satisfy the curve equation over $\mathbb{F}_{q^r}$. Let $\nu = \#E(\mathbb{F}_q)$, the number of points on the curve including ∞. Let p be a prime $\neq \chi$ and $\nmid \chi - 1$, such that $p \mid \nu$ and $p^2 \nmid \nu$. Thus, there exists a subgroup $\mathbb{G}_1'$ of order p in $E(\mathbb{F}_q)$. Let κ be the *embedding degree* of $\mathbb{G}_1'$ in $E(\mathbb{F}_q)$, *i.e.*, the smallest integer ≥ 1 such that $p \mid q^\kappa - 1$, but $p \nmid q^r - 1$ for $1 \leq r \leq \kappa$. Under those conditions, there exist a subgroup $\mathbb{G}_1''$ of order p in $E(\mathbb{F}_{q^\kappa})$, and a subgroup $\mathbb{G}_2$ of order p in the multiplicative group $\mathbb{F}_{q^\kappa}^\star$. For appropriately chosen curves, one can then construct a non-degenerate bilinear map $\mathbf{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ believed to satisfy the BDH assumption, where $\mathbb{G}_1$ is either $\mathbb{G}_1'$ or $\mathbb{G}_1''$.

Specifically, [7] show how to obtain a non-degenerate pairing $\bar{\mathbf{e}} : \mathbb{G}_1' \times \mathbb{G}_1'' \to \mathbb{G}_2$, based on the Tate or the Weil pairing, which can then be combined with a computable isomorphism $\psi : \mathbb{G}_1'' \to \mathbb{G}_1'$, called the *trace map*, to obtain a suitable bilinear map $\mathbf{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ with $\mathbb{G}_1 = \mathbb{G}_1''$. Alternatively, selected curves afford efficiently computable isomorphisms $\phi : \mathbb{G}_1' \to \mathbb{G}_1''$, called *distortion maps*, which can be combined with $\bar{\mathbf{e}}$ to yield pairings of the form $\mathbf{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ with $\mathbb{G}_1 = \mathbb{G}_1'$. The benefit of the latter construction is that the elements of $\mathbb{G}_1'$ have more compact representations than those of $\mathbb{G}_1''$.

It is desired that p and q^κ be large enough for the discrete logarithm to be intractable in generic groups of size p and in the multiplicative group $\mathbb{F}^\star_{q^\kappa}$. Most commonly, q is a large prime or power of 2 or 3, and $\log p \geq 160$, $\log q^\kappa \geq 1000$. We refer the reader to [4] for background information, and to [5] and [14] for details on the concrete implementation.

In the sequel, we treat the above notions as abstract mathematical objects satisfying the properties summarized in Definitions 7, 8, and 9.

Based on this setup, the Boneh-Franklin system defines four operations, the first two for setup and key extraction purposes by the PKG, the last two for encryption and decryption purposes. The two PKG functions are recalled below.

$\mathsf{bfSetup}$ On input a security parameter $n \in \mathbb{N}$: obtain $\langle \mathbb{G}_1, \mathbb{G}_2, p, \mathbf{e} \rangle \leftarrow \mathcal{G}[1^n]$ from the BDH parameter generator; pick two random elements $g \in \mathbb{G}_1^\star$ and $\sigma \in \mathbb{F}_p^\star$, set $g^\sigma = (g)^\sigma \in \mathbb{G}_1^\star$; and construct the hash function $H_0 : \{0,1\}^* \to \mathbb{G}_1^\star$. Finally, output the common public parameters $\pi = \langle \mathbb{G}_1, \mathbb{G}_2, p, \mathbf{e}, g, g^\sigma, H_0 \rangle$ and the master secret $\sigma = \sigma$.

$\mathsf{bfExtract}$ On input $\mathsf{id} \in \{0,1\}^*$: hash the given identity into a public element $i_{\mathsf{id}} = H_0[\mathsf{id}] \in \mathbb{G}_1^\star$, and output $d_{\mathsf{id}} = (i_{\mathsf{id}})^\sigma \in \mathbb{G}_1^\star$ as the private key $\mathsf{pvk}_{\mathsf{id}}$.

5 Encryption-Signature Scheme

We now present an efficient realization of the abstract IBSE specifications of §2.

Table 1 details the six algorithms of our scheme. The Setup and Extract functions are essentially the same as in the original Boneh-Franklin system [5]. Sign and Encrypt implement the IBS of [8], although other randomized signature schemes could be substituted for it. Encrypt and Decrypt are less conventional.

Intuitively, Sign implements a randomized IBS whose signatures comprise a commitment j to some random r chosen by the sender, and a closing v that depends on r and the message m. Encrypt superposes two layers of (expansionless) deterministic encryption. The inner layer encrypts j into x using a minimalist authenticated IBE built from zero-round pairing-based key agreement. The outer layer concurrently determines the value w that encrypts to the same x under a kind of anonymous IBE, derandomized to rely on the entropy already present in x. Then, w is hashed into a one-time pad to encrypt the second half of the signature v, which in turn seeds a one-time pad for the bulk encryption of m.

It is helpful to observe that the exponentiations $\star^r$ and $\star^k$ used in Sign for commitment and in Encrypt for authenticated encryption, as well as the key extraction $\star^\sigma$, and the bilinear pairing $\mathbf{e}[\star, i_B]$ that intervenes in the determination of w, all commute. The legitimate recipient derives its ability to decrypt x from the capacity to perform all of the above operations (either explicitly or implicitly)—but it can only do so in a specific order, different than the sender.

The results of §7 show the scheme is secure. We now prove its consistency.

Theorem 10. *The IBSE scheme of Table 1 is consistent.*

Proof. For decryption, if $\langle \hat{x}, \hat{y}, \hat{z} \rangle = \langle x, y, z \rangle$, it follows that $\hat{w} = \mathbf{e}[{i_A}^{r\,k}, {i_B}^\sigma] = \mathbf{e}[{i_A}^\sigma, i_B]^{r\,k} = w$ (in $\mathbb{G}_2^\star$), and thus $\hat{v} = v$ and $\langle \hat{\mathsf{id}}_A, \hat{m} \rangle = \langle \mathsf{id}_A, m \rangle$; we also have $\hat{u} = \mathbf{e}[\hat{i}_A, i_B]^\sigma = u$ (in $\mathbb{G}_2^\star$), hence $\hat{k} = k$ (in $\mathbb{F}_p^\star$), and thus $\hat{j} = (j^k)^{\hat{k}^{-1}} = j$ (in $\mathbb{G}_1^\star$). For verification, if $\langle \hat{m}, \hat{\mathsf{id}}_A, \hat{j}, \hat{v} \rangle = \langle m, \mathsf{id}_A, j, v \rangle$, we have $\mathbf{e}[g, \hat{v}] = \mathbf{e}[g, i_A]^{\sigma\,(r+h)} = \mathbf{e}[g^\sigma, (\hat{i}_A)^h\,(\hat{i}_A)^r] = \mathbf{e}[g^\sigma, (\hat{i}_A)^h\,\hat{j}]$ (in $\mathbb{G}_2$), as required. $\square$

Table 1: The IBSE algorithms. The hash functions are modeled as random oracles. The output of H_4 is viewed as a stream that is truncated as dictated by context, *viz.*, $H_4[\mathsf{key}] \oplus \mathsf{data}$ perfoms a length-preserving "one-time pad" encryption or decryption.

Setup On input a security parameter $n \in \mathbb{N}$: establish the Boneh-Franklin parameters $\mathbb{G}_1$, $\mathbb{G}_2$, p, $\mathbf{e}$, g, g^σ, σ as in bfSetup, and select five hash functions $H_0 : \{0,1\}^* \to \mathbb{G}_1^\star$, $H_1 : \mathbb{G}_1^\star \times \{0,1\}^* \to \mathbb{F}_p^\star$, $H_2 : \mathbb{G}_2^\star \to \{0,1\}^{\lceil \log p \rceil}$, $H_3 : \mathbb{G}_2^\star \to \mathbb{F}_p^\star$, $H_4 : \mathbb{G}_1 \to \{0,1\}^*$; then, output the common public parameters $\langle \mathbb{G}_1, \mathbb{G}_2, p, \mathbf{e}, g, g^\sigma, H_0, H_1, H_2, H_3, H_4 \rangle$ and the master secret σ.

Extract On input $\mathsf{id} \in \{0,1\}^*$: proceed as in bfExtract.

Sign On input the private key d_A of some sender identity id_A, and a message m:
derive $i_A = H_0[\mathsf{id}_A]$ (so $d_A = (i_A)^\sigma$),
pick a random $r \in \mathbb{F}_p^\star$,
let $j = (i_A)^r \in \mathbb{G}_1^\star$,
let $h = H_1[j, m] \in \mathbb{F}_p^\star$,
let $v = (d_A)^{r+h} \in \mathbb{G}_1$,
then, output the signature $\langle j, v \rangle$; also forward $\langle m, r, \mathsf{id}_A, i_A, d_A \rangle$ for further use by Encrypt.

Encrypt On input a recipient identity id_B, and $\langle j, v, m, r, \mathsf{id}_A, i_A, d_A \rangle$ from Sign as above:
derive $i_B = H_0[\mathsf{id}_B]$,
compute $u = \mathbf{e}[d_A, i_B] \in \mathbb{G}_2^\star$,
let $k = H_3[u] \in \mathbb{F}_p^\star$;
set $x = j^k \in \mathbb{G}_1^\star$,
let $w = u^{k\,r} \in \mathbb{G}_2^\star$,
set $y = H_2[w] \oplus v$,
set $z = H_4[v] \oplus \langle \mathsf{id}_A, m \rangle$;
then, output the ciphertext $\langle x, y, z \rangle$.

Decrypt On input a private key d_B for id_B, and an anonymous ciphertext $\langle \hat{x}, \hat{y}, \hat{z} \rangle$:
derive $i_B = H_0[\mathsf{id}_B]$,
compute $\hat{w} = \mathbf{e}[\hat{x}, d_B]$,
recover $\hat{v} = H_2[\hat{w}] \oplus \hat{y}$,
recover $\langle \hat{\mathsf{id}}_A, \hat{m} \rangle = H_4[\hat{v}] \oplus \hat{z}$,
derive $\hat{i}_A = H_0[\hat{\mathsf{id}}_A]$,
compute $\hat{u} = \mathbf{e}[\hat{i}_A, d_B]$,
let $\hat{k} = H_3[\hat{u}]$,
set $\hat{j} = \hat{x}^{\hat{k}^{-1}}$;
then, output the decrypted message $\hat{m}$, the signature $\langle \hat{j}, \hat{v} \rangle$, and the purported identity of the originator $\hat{\mathsf{id}}_A$.

Verify On input a signed message $\langle \hat{m}, \hat{j}, \hat{v} \rangle$ by purported sender identity $\hat{\mathsf{id}}_A$:
derive $\hat{i}_A = H_0[\hat{\mathsf{id}}_A]$,
let $\hat{h} = H_1[\hat{j}, \hat{m}]$,
check whether $\mathbf{e}[g, \hat{v}] \stackrel{?}{=} \mathbf{e}[g^\sigma, (\hat{i}_A)^{\hat{h}}\, \hat{j}]$;
then, output $\top$ if the equality holds, output $\perp$ otherwise.

---

6 Competitive Performance

Table 2 gives a comparison between various IB encryption and signature schemes, in terms of size, performance, and security properties.

Our comparisons include most relevant pairing-based IB schemes for encryption, authenticated encryption, signature, and signcryption. We also include a suite of hybrid schemes, obtained by combining IBS [8] with either IBE [5] or AuthIBE [20]; each pair is composed in three different ways depending on the order of application of the primitives: encrypt-then-sign ($\mathcal{E}t\mathcal{S}$), sign-then-encrypt ($\mathcal{S}t\mathcal{E}$), and commit-then-parallel-encrypt-and-sign ($\mathcal{C}t\mathcal{S}\&\mathcal{E}$), as per [1]. Roughly speaking, in $\mathcal{C}t\mathcal{S}\&\mathcal{E}$, the plaintext m is reversibly transformed into a redundant pair $\langle a, b \rangle$, where a is a commitment to m that reveals "no information" about m; then, a is signed and b encrypted using the given primitives, in parallel.

For fairness, the size comparison factors out the overhead of explicitly including the sender identity to the signed plaintext prior to encryption; our scheme does this to avoid sending the

Table 2: Comparison between various IB encryption, signature, signcryption, and multipurpose schemes. Times are expressed as triples $\langle \#b, \#m, \#e \rangle$, where $\#b$ is the number of bilinear pairings, $\#m$ is the number of exponentiations in $\mathbb{G}_1$, and $\#e$ is the number of exponentiations in $\mathbb{G}_2$ (simple group operations in $\mathbb{G}_1$ and multiplications and inversions in $\mathbb{F}_p$ or $\mathbb{G}_2$ are omitted). Sizes are reported as pairs $\langle \#p, \#q \rangle$, where $\#p$ is the number of $\mathbb{G}_1$ elements, and $\#q$ is the number of $\mathbb{F}_p$ elements, in excess of the original unsigned message size $\|m\|$ taken as baseline (treating the sender identity as part of m, if included); the 'cipher' size for schemes that support encryption is the ciphertext overhead, given by $\|c\| - \|m\|$, whereas the 'plain' size for schemes that support plaintext signatures is the signature overhead after decryption, given by $\|\langle m, s\rangle\| - \|m\|$. Supported security properties are indicated as follows: message Confidentiality, signature Non-repudiation, ciphertext Authentication, ciphertext Unlinkability, and ciphertext anOnymity; for non-IBSE schemes, an uppercase denotes an analogously strong security notion, a lowercase a weaker notion; a star means a broken scheme.

Scheme	**Security**: Conf, Nrep,Auth,Ulnk,anOn	**Size**: #elt.$\mathbb{G}_1$, $\mathbb{F}_p$		**Time**: #pairings, #exp.$\mathbb{G}_1$, #exp.$\mathbb{G}_2$			
		Cipher	Plain	Sign	Encrypt	Decrypt	Verify
IB Encryption [5]	C,–, –, U,O	1, 1	—	—	1, 0, 0	1, 1, 0	—
IB Auth.Encr. [20]	C,–, A,U,–	0, 2	—	—	1, 0, 0	1, 0, 0	—
IB Signature [8]	–, N,A,–, –	—	2, 0	0, 2, 0	—	—	2, 1, 0
[a] IB Signature [22]	–, N,A,–, –	—	2, 0	0, 4, 0	—	—	2(3), 0, 2
IB Sign. [16, #3]	–, N,A,–, –	—	1, 1	1, 2, 1	—	—	2, 0, 1
IB Sign. [16, #4]	–, N,A,–, –	—	2, 0	0, 2, 0	—	—	2, 0, 1
[b] IB SignCrypt. [21]	*, N,A,–, –	2, 0	2, 0	··· 1, 3, 0 ···		··· 4, 0, 1 ···	
IB SignCrypt. [19]	C,N,A,–, –	1, 1	1, 1	··· 2, 2, 0 ···		··· 4, 1, 0 ···	
[c] IB E–then–S	c, N,A,–, –	3, 1	—	0, 2, 0	1, 0, 0	1, 1, 0	2, 1, 0
[c] IB S–then–E	C,n, –, U,O	3, 1	2, 0	0, 2, 0	1, 0, 0	1, 1, 0	2, 1, 0
[c] IB commit–E&S	C,N,A,U,–	3, 1, +	2, 0, +	0, 2, 0	1, 0, 0	1, 1, 0	2, 1, 0
[d] IB AE–then–S	c, N,A,U,–	2, 2	—	0, 2, 0	1, 0, 0	1, 0, 0	2, 1, 0
[d] IB S–then–AE	C,n, A,U,–	2, 2	2, 0	0, 2, 0	1, 0, 0	1, 0, 0	2, 1, 0
[d] IB commit–AE&S	C,N,A,U,–	2, 2, +	2, 0, +	0, 2, 0	1, 0, 0	1, 0, 0	2, 1, 0
IBSE: this paper	C,N,A,U,O	2, 0	2, 0	0, 2, 0	1, 1, 1	2, 1, 0	2, 1, 0

[a] Signature verification in [22] requires 3 pairings, one of which may be precomputed.

[b] The signcryption scheme of [21] is not adaptive CCA-secure, see [19] for details.

[c] These are compositions of IBE [5] and IBS [8, 16] using $\mathcal{E}t\mathcal{S}$, $\mathcal{S}t\mathcal{E}$, $\mathcal{C}t\mathcal{S}\&\mathcal{E}$ from [1].

[d] These are compositions of AuthIBE [20] and IBS [8, 16] using $\mathcal{E}t\mathcal{S}$, $\mathcal{S}t\mathcal{E}$, $\mathcal{C}t\mathcal{S}\&\mathcal{E}$ [1].

Notes on hybrid schemes ([c],[d]): $\mathcal{E}t\mathcal{S}$ and $\mathcal{S}t\mathcal{E}$ only support degraded notions of CCA indistinguishability and CMA unforgeability in the insider model, respectively, hence the lowercase 'c' and 'n'; for $\mathcal{C}t\mathcal{S}\&\mathcal{E}$, the '+' serves to remind that this more secure composition incurs extra overhead due to the commitment redundancy. See [1] for details.

identity in the clear. Note that all authenticated communication schemes require the recipient to get hold of that information, but most simply assume that it is conveyed using a different channel.

Evidently, the proposed scheme offers an interesting solution to the problem of identity-based signed encryption: it offers an unmatched combination of security features that not only provide the usual confidentiality/non-repudiation requirements, but also guarantee authentication, anonymity, and unlinkability of the ciphertext. Our scheme achieves all this at a cost comparable to that of monolithic IB signcryption, and in a significantly tighter package than any generic combination of existing IB encryption and signature algorithms.

By comparison, the two listed signcryption schemes have comparable spatial and computational overheads but, by the very nature of monolithic signcryption, cannot offer ciphertext anonymity.

As for the suite of generic compositions, they have a slight advantage in terms of cost, but incur a large size penalty, and require us to choose between ciphertext authentication and anonymity.

We also note that, in the original Boneh-Franklin setup, the IBSE ciphertexts and signed plaintexts are essentially as compact as that of IBE or IBS taken in isolation; this is generally true when $p \approx q$, and when $\mathbb{G}_1 = \mathbb{G}'_1$ so that its points can be represented as elements of $\mathbb{F}_q$ using point compression [4]. However, the schemes of [5], [19], and especially [20] have smaller ciphertexts and signatures, as the case may be, in generalized setups where $p \ll q$, or $\mathbb{G}_1 = \mathbb{G}''_1$.

7 Security Analysis

We now state our security results for the scheme of §5 in the models of §3. These results are given under the irreflexivity assumption[1].

Theorem 11. *Let $\mathcal{A}$ be a polynomial-time IND-IBSE-CCA attacker that has advantage $\geq \epsilon$, and makes $\leq \mu_i$ queries to the random oracles H_i, $i = 0, 1, 2, 3, 4$. Then, there exists a polynomial-time algorithm $\mathcal{B}$ that solves the bilinear Diffie-Hellman problem with advantage $\geq \epsilon/(\mu_0\, \mu_2)$.*

Theorem 12. *Let $\mathcal{A}$ be an EUF-IBSE-CCA attacker that makes $\leq \mu_i$ queries to the random oracles H_i, $i = 0, 1, 2, 3, 4$, and $\leq \mu_{\rm se}$ queries to the signature/encryption oracle. Assume that, within a time span $\leq \tau$, $\mathcal{A}$ produces a successful forgery with probability $\geq \epsilon = 10\,(\mu_{\rm se} + 1)\,(\mu_{\rm se} + \mu_1)/2^n$, for a security parameter n. Then, there exists an algorithm $\mathcal{B}$ that solves the bilinear Diffie-Hellman problem in expected time $\leq 120686\, \mu_0\, \mu_1\, \tau/\epsilon$.*

Theorem 13. *There exists a polynomial-time algorithm that, given an identifier id_A, a signed plaintext $\langle m, j, v\rangle$ from id_A, and a private key d_B, creates a ciphertext $\langle x, y, z\rangle$ that decrypts to $\langle m, j, v\rangle$ under d_B, with probability 1.*

Theorem 14. *Let $\mathcal{A}$ be a polynomial-time AUTH-IBSE-CMA attacker with advantage $\geq \epsilon$, that makes $\leq \mu_i$ queries to the random oracles H_i, $i = 0, 1, 2, 3, 4$. Then, there exists a polynomial-time algorithm $\mathcal{B}$ that solves the bilinear D-H problem with advantage $\geq 2\,\epsilon/(\mu_0\,(\mu_0 - 1)\,(\mu_1\, \mu_2 + \mu_3))$.*

Theorem 15. *Let $\mathcal{A}$ be a polynomial-time ANON-IBSE-CCA attacker that has advantage $\geq \epsilon$, and makes $\leq \mu_i$ queries to the random oracles H_i, $i = 0, 1, 2, 3, 4$. Then, there exists a polynomial-time algorithm $\mathcal{B}$ that solves the bilinear Diffie-Hellman problem with advantage $\geq 3\,\epsilon/(\mu_0\,(\mu_0 - 1)\,(\mu_1\, \mu_2 + 2\, \mu_2 + \mu_3))$.*

8 Practical Extensions

We now mention a few straightforward generalizations of practical interest.

8.1 Encrypting for Multiple Recipients

Encrypting the same message m for a set of n recipients $\mathsf{id}_{B_1}, \ldots, \mathsf{id}_{B_n}$ is easily achieved as follows. The Sign operation is carried out once (which establishes the randomization parameter r), then the $\mathsf{Encrypt}$ operation is performed independently for each recipient, based on the output from Sign.

[1] Analogous claims can be made in the general case without the irreflexivity assumption, *i.e.*, allowing the oracles to encrypt and decrypt messages with the same sender and recipient keys; however the reductions from the BDH problem are significantly more involved and less efficient. See Appendix D for details.

Since the message m and the randomization parameter r are invariant for all the $\mathsf{Encrypt}$ instances, it is easy to see that the z component of the ciphertext also remains the same. Thus, the multi-recipient composite ciphertext is easily assembled from one instance of $\langle x_i, y_i \rangle \in \mathbb{G}_1^\star \times \mathbb{G}_1^\star$ for each recipient B_i, plus a single instance of $z \in \{0,1\}^*$ to be shared by all. Thus, a multi-recipient ciphertext is compactly encoded in the form $c = \langle \langle x_1, y_1 \rangle, ..., \langle x_n, y_n \rangle, z \rangle$.

The security models of §3 have to support two additional types of queries: *multi-recipient signature/encryption queries*, in which a given message, sender, and list of recipients, are turned into a multi-recipient ciphertext, and *multi-recipient decryption queries*, in which the individual elements of a multi-recipient ciphertext are decrypted, under a given identity, and a valid plaintext is returned, if there is any. The modified security analysis is deferred to the full paper[2].

8.2 Integrity Without Non-Repudiation

The scheme of §5 is trivially modified to provide message integrity without non-repudiation or authentication. To do this, the sender merely substitutes the public parameters $\langle g, g^\sigma \rangle$ for $\langle i_A, d_A \rangle$, wherever the sender's key pair is used in the Sign and $\mathsf{Encrypt}$ operations. The sender also tags the message as 'anonymous', instead of specifying an identity. Similarly, the $\mathsf{Decrypt}$ and Verify operations are performed substituting g for $\hat{i}_A$ wherever it appears as a function argument.

This is valid since the key pair relation $d_A = (i_A)^\sigma$ is paralleled by $g^\sigma = (g)^\sigma$, but authentication is meaningless since the signing 'private' key g^σ is public.

9 Conclusion

In this paper, we have proposed a comprehensive security model for multi-purpose identity-based encryption-signature cryptosystems. Our security model defines five core properties that we believe precisely capture what a consumer of cryptography intuitively expects when he or she wishes to engage in "secure signed communication" with a remote party. It bears repeating that these do not only include the standard confidentiality and non-repudiation requirements, but also the much less commonly offered features of ciphertext authentication, ciphertext deniability or unlinkability, and true ciphertext anonymity with respect to third parties. We have given precise definitions for all these properties in the context of identity-based cryptography.

As second contribution, we have presented a new cryptographic scheme that precisely implements all facets of the aforementioned notion of "secure signed communication", in the certificate-free world of identity-based cryptography. Our scheme offers efficient security bounds in all the above respects; it is fast, compact, scalable, and practical—as we have illustrated through detailed comparisons with most or all mainstream identity-based cryptosystems to date.

Acknowledgements

The author would like to thank Dan Boneh, Jonathan Katz, and the anonymous referees of Crypto 2003 for many helpful suggestions and comments. Credit goes to Guido Appenzeller for suggesting the "Swiss Army Knife" moniker.

[2]The corresponding proofs of security are straightforward generalizations of the single-recipient case. See Appendix C for details.

References

[1] J.H. An, Y. Dodis, and T. Rabin. On the security of joint signature and encryption. In *Proc. Eurocrypt '02, LNCS 2332*, 2002.

[2] P.S.L.M. Barreto, H.Y. Kim, B. Lynn, and M. Scott. Efficient algorithms for pairing-based cryptosystems. In *Proc. Crypto '02, LNCS 2442*, 2002.

[3] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Proc. Conf. Computer and Communication Security*, 1993.

[4] I. Blake, G. Seroussi, and N. Smart. *Elliptic Curves in Cryptography.* Cambridge University Press, 1999.

[5] D. Boneh and M. Franklin. Identity-based encryption from the Weil pairing. In *Proc. Crypto '01, LNCS 2139*, pages 213–229, 2001. See [6] for the full version.

[6] D. Boneh and M. Franklin. Identity based encryption from the weil pairing. Cryptology ePrint Archive, Report 2001/090, 2001. http://eprint.iacr.org/.

[7] D. Boneh, B. Lynn, and H. Shacham. Short signatures from the Weil pairing. In *Proc. Asiacrypt '01, LNCS 2248*, pages 514–532, 2001.

[8] J.C. Cha and J.H. Cheon. An identity-based signature from gap Diffie-Hellman groups. Cryptology ePrint Archive, Report 2002/018, 2002. http://eprint.iacr.org/.

[9] L. Chen and C. Kudla. Identity based authenticated key agreement from pairings. Cryptology ePrint Archive, Report 2002/184, 2002. http://eprint.iacr.org/.

[10] C. Cocks. An identity based encryption scheme based on quadratic residues. In *Proc. 8th IMA Int. Conf. Cryptography and Coding*, pages 26–28, 2001.

[11] U. Feige, A. Fiat, and A. Shamir. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM J. Computing*, 17(2):281–308, 1988.

[12] U. Feige, A. Fiat, and A. Shamir. Zero-knowledge proofs of identity. *J. Cryptology*, 1:77–94, 1988.

[13] A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Proc. Crypto '86, LNCS 263*, pages 186–194, 1984.

[14] S.D. Galbraith, K. Harrison, and D. Soldera. Implementing the Tate pairing. Technical Report HPL-2002-23, HP Laboratories Bristol, 2002.

[15] C. Gentry and A. Silverberg. Hierarchical ID-based cryptography. Cryptology ePrint Archive, Report 2002/056, 2002. http://eprint.iacr.org/.

[16] F. Hess. Exponent group signature schemes and efficient identity based signature schemes based on pairings. Cryptology ePrint Archive, Report 2002/012, 2002. http://eprint.iacr.org/.

[17] J. Horwitz and B. Lynn. Toward hierarchical identity-based encryption. In *Proc. Eurocrypt '02, LNCS 2332*, pages 466–481, 2002.

[18] A. Joux. A one round protocol for tripartite Diffie-Hellman. In *Proc. 4th Alg. Numb. Th. Symp., LNCS 1838*, pages 385–294, 2000.

[19] B. Libert and J.-J. Quisquater. New identity based signcryption schemes based on pairings. Cryptology ePrint Archive, Report 2003/023, 2003. http://eprint.iacr.org/.

[20] B. Lynn. Authenticated identity-based encryption. Cryptology ePrint Archive, Report 2002/072, 2002. http://eprint.iacr.org/.

[21] J. Malone-Lee. Identity-based signcryption. Cryptology ePrint Archive, Report 2002/098, 2002. http://eprint.iacr.org/.

[22] K.G. Paterson. ID-based signatures from pairings on elliptic curves. Cryptology ePrint Archive, Report 2002/004, 2002. http://eprint.iacr.org/.

[23] R. Sakai, K. Ohgishi, and M. Kasahara. Cryptosystems based on pairings. In *Proc. SCIS '00*, pages 26–28, Okinawa, Japan, 2000.

[24] A. Shamir. Identity-based cryptosystems and signature schemes. In *Proc. Crypto '84, LNCS 196*, pages 47–53, 1984.

[25] N.P. Smart. An identity based authenticated key agreement protocol based on the Weil pairing. Cryptology ePrint Archive, Report 2001/111, 2001. http://eprint.iacr.org/.

[26] D. Pointcheval J. Stern. Security arguments for digital signatures and blind signatures. *J. Cryptology*, 13:361–396, 2000.

[27] Y. Zheng. Digital signcryption or how to achieve *cost* (*signature* & *encryption*) $\ll$ *cost* (*signature*) + *cost* (*encryption*). In *Proc. Crypto '97, LNCS 1294*, 1997.

A Related IBE and IBS Schemes

A.1 The Boneh-Franklin IBE

In this section, we give a brief description of the Boneh-Franklin IBE for a fixed message size κ, as originally described in [5]. The following two functions $\mathsf{ibeSetup}$ and $\mathsf{ibeExtract}$ are used by the trusted Private Key Generator (PKG).

$\mathsf{ibeSetup}$ On input a security parameter $n \in \mathbb{N}$: obtain $\langle \mathbb{G}_1, \mathbb{G}_2, p, \mathbf{e}, g, g^\sigma, H_0 \rangle$ as in $\mathsf{bfSetup}$, select a message size $\kappa \in \mathbb{N}$ in function of the security parameter, and construct the additional three hash functions $H'_1 : \mathbb{G}_2 \to \{0,1\}^\kappa$, $H'_2 : \{0,1\}^\kappa \times \{0,1\}^\kappa \to \mathbb{F}_p^\star$, $H'_3 : \{0,1\}^\kappa \to \{0,1\}^\kappa$. Finally, output the common public parameters $\pi = \langle \mathbb{G}_1, \mathbb{G}_2, p, \mathbf{e}, g, g^\sigma, \kappa, H_0, H'_1, H'_2, H'_3 \rangle$ and the master secret $\sigma = \sigma$. (Note: 'BasicIBE' uses only H_0 and H'_1; 'FullIBE' also uses H'_2 and H'_3.)

$\mathsf{ibeExtract}$ On input $\mathsf{id} \in \{0,1\}^*$: proceed as $\mathsf{bfExtract}$.

The next two functions $\mathsf{ibeBasicEncrypt}$ and $\mathsf{ibeBasicDecrypt}$ implement the first 'basic' IBE described in [5, §4.1], which offers semantic security against passive attacks.

$\mathsf{ibeBasicEncrypt}$ On input a message $m \in \{0,1\}^\kappa$, and a recipient identity $\mathsf{id} \in \{0,1\}^*$: derive the public element $i_{\mathsf{id}} = H_0[\mathsf{id}]$, pick a random number $r \in \mathbb{F}_p^\star$, calculate $w_{\mathsf{id}} = \mathbf{e}[i_{\mathsf{id}}, g^\sigma] \in \mathbb{G}_2^\star$, and output $\langle g^r, H'_1[w_{\mathsf{id}}^r] \oplus m \rangle$ as the ciphertext c.

$\mathsf{ibeBasicDecrypt}$ On input a received ciphertext $\hat{c} = \langle \hat{c}_1, \hat{c}_2 \rangle$, and a private key $d_{\mathsf{id}} \in \mathbb{G}_1^\star$: compute and return $H'_1[\mathbf{e}[d_{\mathsf{id}}, \hat{c}_1]] \oplus \hat{c}_2$ as the decrypted message $\hat{m}$.

The last two functions $\mathsf{ibeFullEncrypt}$ and $\mathsf{ibeFullDecrypt}$ implement a Fujisaki-Okamoto strengthening of the basic IBE above into the full IBE of [5, §4.2], which is semantically secure against CCA attacks.

$\mathsf{ibeFullEncrypt}$ On input a message $m \in \{0,1\}^\kappa$, and a recipient identity $\mathsf{id} \in \{0,1\}^*$: derive $i_{\mathsf{id}} = H_0[\mathsf{id}]$, pick a random number $t \in \mathbb{F}_p^\star$, set $r = H'_2[t, m]$, calculate $w_{\mathsf{id}} = \mathbf{e}[i_{\mathsf{id}}, g^\sigma] \in \mathbb{G}_2^\star$, and output $\langle g^r, H'_1[w_{\mathsf{id}}^r] \oplus t, H'_3[t] \oplus m \rangle$ as the ciphertext c.

$\mathsf{ibeFullDecrypt}$ On input a received ciphertext $\hat{c} = \langle \hat{c}_1, \hat{c}_2, \hat{c}_3 \rangle$, and a private key $d_{\mathsf{id}} \in \mathbb{G}_1^\star$: compute $\hat{t} = H'_1[\mathbf{e}[d_{\mathsf{id}}, \hat{c}_1]] \oplus \hat{c}_2$, compute $\hat{m} = H'_3[t] \oplus \hat{c}_3$, and let $\hat{r} = H'_2[\hat{t}, \hat{m}]$; if $g^{\hat{r}} \neq \hat{c}_1$, reject the ciphertext, otherwise, output $\hat{m}$ as the decrypted message.

A.2 The Cha-Cheon IBS

We now briefly describe one of several comparable IB signature schemes inspired from the BF IBE [22, 16, 8]. The following is the Cha-Cheon scheme [8] for signing messages in $\{0,1\}^*$.

$\mathsf{ibsSetup}$ On input $n \in \mathbb{N}$, do as $\mathsf{bfSetup}$, and construct an additional hash function function $H''_1 : \{0,1\}^* \times \mathbb{G}_1 \to \mathbb{F}_p^\star$.

$\mathsf{ibsExtract}$ On input $\mathsf{id} \in \{0,1\}^*$, do as $\mathsf{bfExtract}$.

$\mathsf{ibsSign}$ On input a message $m \in \{0,1\}^*$, an identity id, and the private key $d_{\mathsf{id}} \in \mathbb{G}_1^\star$: derive $i_{\mathsf{id}} = H_0[\mathsf{id}]$, pick a random number $r \in \mathbb{F}_p^\star$, let $t = (i_{\mathsf{id}})^r$, $h = H''_1[m, t]$, $v = (d_{\mathsf{id}})^{t+h}$, and output the pair $\langle t, v \rangle$ as the signature s.

$\mathsf{ibsVerify}$ On input an identity id, a message $\hat{m}$, and a signature $\langle \hat{t}, \hat{v} \rangle$: derive $i_{\mathsf{id}} = H_0[\mathsf{id}]$, compute $\hat{h} = H''_1[\hat{m}, \hat{t}]$, and verify that $\mathbf{e}[g^\sigma, (i_{\mathsf{id}})^{\hat{h}}\, \hat{t}] = \mathbf{e}[g, \hat{v}]$.

B Proofs of Security

The proofs of four out of the five security theorems (Theorems 11, 12, 14, and 15) use a reduction argument based on a simulation. Since the simulations are fairly similar amongst the proofs, the common aspects are captured in a generic simulator described in a lemma (Lemma 17). The generic simulator is constructed to make extensive use of the random oracles, but leaves the final reduction open for customization.

The remaining theorem (Theorem 13) is shown using the direct construction of an additional feature of the IBSE system, also used in other proofs, the gist of which is presented in another lemma (Lemma 16).

We begin by stating and proving Lemmas 16 and 17.

Lemma 16. *There exists an efficient algorithm* $\mathsf{EncryptToSelf}$ *that, given the private key* d_B *of a recipient identity* id_B*, and a valid signed plaintext* $\langle \mathsf{id}_A, m, j, v\rangle$*, encrypts the given plaintext to the given recipient, without being given the private key of* id_A*. The output of* $\mathsf{EncryptToSelf}$ *is indistinguishable from that of* $\mathsf{Encrypt}$*, i.e., the encryption by* $\mathsf{EncryptToSelf}$ *of* $\langle \mathsf{id}_A, m, j, v\rangle$ *under* d_B *has the same distribution as the output of* $\mathsf{Encrypt}$ *on inputs* id_B *and* $\langle j, v, m, r, \mathsf{id}_A, i_A, d_A\rangle$*, when the latter tuple is any output from* Sign *that matches the given values of* $\langle \mathsf{id}_A, m, j, v\rangle$*.*

Proof. Consider the following (polynomial-time, deterministic) algorithm:

$\mathsf{EncryptToSelf}$ On input a signed plaintext $\langle \mathsf{id}_A, m, j, v\rangle$, and a private key d_B:
derive $i_A = H_0[\mathsf{id}_A]$,
compute $u = \mathbf{e}[i_A, d_B]$,
let $k = H_3[u]$,
set $x = j^k$,
compute $w = \mathbf{e}[x, d_B]$,
set $y = H_2[w] \oplus v$,
set $z = H_4[v] \oplus \langle \mathsf{id}_A, m\rangle$;
then, output $\langle x, y, z\rangle$ as the ciphertext c.

The $\mathsf{EncryptToSelf}$ algorithm is easily observed to produce a valid ciphertext from id_A to id_B, provided that the given plaintext was validly signed by id_A.

Since any valid signed plaintext $\langle \mathsf{id}_A, m, j, v\rangle$ corresponds to exactly one legitimate Sign output tuple $\langle j, v, m, r, \mathsf{id}_A, i_A, d_A\rangle$, and, in turn, has exactly one ciphertext under $\mathsf{Encrypt}$ for any given recipient, it follows that $\mathsf{EncryptToSelf}$ will output the same ciphertext for matching values of $\langle \mathsf{id}_A, m, j, v\rangle$.

□

Lemma 17. *There exists an efficient algorithm* $\mathcal{S}$ *that, given a pair of* $\mathbb{G}_1^\star$*-elements* $\langle g, g^\zeta\rangle$*—called 'public'—, and a finite set of* $\mathbb{G}_1^\star$*-elements* $\{g^{\xi_i}\}$*—called 'crucial'—, provides an interactive simulation of all functions of an IBSE oracle with public parameters* $\langle g, g^\zeta\rangle$*, when the queries are subject to the following constraints:*

1. *Never during the course of the simulation, should* $\mathcal{S}$ *be asked to perform either:*
   (a) *private key extraction queries for crucial identities, i.e., any identity* id *whose public key* $H_0[\mathsf{id}]$ *belongs to the set of crucial elements* $\{g^{\xi_i}\}$*;*
   (b) *decryption queries on ciphertexts signed by and encrypted for crucial identities, viz., the oracle output is unspecified on such inputs*[3]*;*
   (c) *signature/encryption queries where the sender and recipient are both crucial.*
2. *At arbitrary times for as many as* $\#\{g^{\xi_i}\}$ *occurences throughout the simulation,* $\mathcal{S}$ *may be directed (by some controlling entity) to evaluate the next previously unseen random oracle query* $H_0[\mathsf{id}]$ *(called a* designated *query) to any previously unused crucial element* $g^\xi \in \{g^{\xi_i}\}$ *(thereby forcibly assigning to the designated identity* id *the crucial public key* g^ξ*).*

[3]It is noted that a *bona fide* adversary interacting with the simulator $\mathcal{S}$ may unwittingly violate Condition 1b in Lemma 17, as the signing identity of a ciphertext is generally concealed until the ciphertext is decrypted. The lemma states that the simulator will produce an unspecified output in such a situation, but will resume normal operation on the subsequent queries.

During the course of a polynomial-length interaction under those conditions, $\mathcal{S}$ will be indistinguishable from a real IBSE oracle (except with some negligible probability that corresponds to the occurence of a random oracle anomaly, i.e., either a collision or the guessing of a preimage).

Proof. We exhibit a construction of $\mathcal{S}$ that we show satisfies the listed requirements.

CONSTRUCTION:

For simplicity, we assume that all random oracle queries made to $\mathcal{S}$ are distinct, and that any query involving an identifier id is preceded by the random oracle query $H_0[\mathsf{id}]$. (This can always be enforced by appropriate caching of the queries.)

In a preliminary step, $\mathcal{S}$ starts by initializing one empty list L_i for each random oracle H_i. $\mathcal{S}$ also maintains a pool of unused crucial elements, initially equal to the entire set of crucial elements.

The various oracle queries are then serviced by $\mathcal{S}$ as follows. As regards queries to the random oracles:

- Queries on H_0 for a given identifier id are handled as follows:
  - If the query at hand is one of the designated queries, then $\mathcal{S}$ fetches a crucial element g^ξ from the unused pool (or verifies that the crucial element designated by the controlling entity is still unused), removes g^ξ from the pool, adds the tuple $\langle \mathsf{id}, g^\xi \rangle$ to L_0, and returns g^ξ. In the sequel, we call any identity determined in this manned a 'guessed' identity, and denote it by id_ξ as a reminder of its newly assigned crucial public key g^ξ. (Note that the exponent ξ is merely a notational expedient; its value is unknown to $\mathcal{S}$.)
  - Otherwise, $\mathcal{S}$ picks a random $\lambda \in \mathbb{F}_p^\star$, add the tuple $\langle \mathsf{id}, \lambda \rangle$ to the list L_0, and return the public key $i = g^\lambda$.
- Queries on H_1, H_2, H_3, and H_4 are handled the obvious way, by producing a randomly sampled element from the appropriate codomain, and adding both query and answer to L_1, L_2, L_3, or L_4, respectively. [4]

As regards oracle queries to $\mathcal{S}$ that involve IBSE operations:

- Signature/encryption queries: suppose $\mathcal{S}$ is asked to sign a message m in the name of sender id_A and encrypt it for recipient id_B.
  - If id_A is some guessed identity id_ξ, then $\mathcal{S}$ proceeds as follows. First, $\mathcal{S}$ picks two random numbers $r, h \in \mathbb{F}_p^\star$, lets $j = g^r\,(g^\xi)^{-h}$, and $v = (g^\zeta)^r$. Next, $\mathcal{S}$ adds the tuple $\langle j, m, h \rangle$ to the list L_1 (thereby forcing the value of the simulated random oracle to $H_1[j, m] = h$). Hence, $\mathcal{S}$ now has a signature $\langle j, v \rangle$ for the given message m and signer identity id_ξ. Then:
    - If id_B is also a guessed identity, *i.e.*, both the sender and the recipient have crucial public keys, then $\mathcal{S}$ signals a violation of the assumptions.
    - Otherwise, $\mathcal{S}$ computes the private key of id_B, which is given by $d_B = (g^\zeta)^{\lambda_B}$, where λ_B is found in L_0. The last step is performed by applying the function $\mathsf{EncryptToSelf}$ from Lemma 16 to the signed message $\langle \mathsf{id}_\xi, m, j, v \rangle$ and the private key d_B, which produces the desired ciphertext $\langle x, y, z \rangle$.
  - If instead id_A is not a guessed identity, then $\mathcal{S}$ computes the private key $d_A = (g^\zeta)^{\lambda_A}$ of the sender id_A, where λ_A is found in L_0, and computes the desired ciphertext the regular way using Sign followed by $\mathsf{Encrypt}$.
- Decryption queries: suppose $\mathcal{S}$ is given a recipient identity id_B and a ciphertext $c = \langle x, y, z \rangle$.

[4]Note regarding the simulation of H_4: we have assumed that H_4 produced an output stream of unbounded length, that was then truncated as needed (*i.e.*, to match the size of the data to encrypt, see Table 1). In this proof and all the others, we assume for simplicity that the output of H_4 has a definite length that is known from context. It would be tedious but straightforward to simulate H_4 as a stream in all generality and maintain L_4 accordingly.

- If id_B is some guessed identity id_ξ, then $\mathcal{S}$ does the following. First, the lists are searched for all combinations $\langle \mathsf{id}_A, m, j, v\rangle$ such that $\langle j, m, h_1\rangle \in L_1$, $\langle w, h_2\rangle \in L_2$, $\langle u, h_3\rangle \in L_3$, $\langle v, h_4\rangle \in L_4$, for some h_1, h_2, h_3, h_4, u, w, under the constraints that $h_2 \oplus y = v$, $x^{h_3^{-1}} = j$, $h_4 \oplus z = \langle \mathsf{id}_A, m\rangle$, and $\mathsf{Verify}[\mathsf{id}_A, m, j, v] = \top$. Then, for each such $\langle \mathsf{id}_A, m, j, v\rangle$:
  * If id_A is also a guessed identity, then that particular instance $\langle \mathsf{id}_A, m, j, v\rangle$ is rejected from further consideration as a decryption candidate.
  * Otherwise, if the two additional constraints that $\mathbf{e}[v\,(d_A)^{-h_1}, g^\xi]^{h_3} = w$ and $\mathbf{e}[d_A, g^\xi] = u$ are satisfied, where $d_A = (g^\zeta)^{\lambda_A}$ is the private key of id_A, with λ_A such that $\langle \mathsf{id}_A, \lambda_A\rangle \in L_0$, then the tuple $\langle \mathsf{id}_A, m, j, v\rangle$ is selected as a decryption candidate.

  Finally, the oracle selects a decryption candidate $\langle \mathsf{id}_A, m, j, v\rangle$ that satisfies all the above conditions (breaking ties arbitrarily), and returns it as the decrypted signed plaintext for the query. In case no eligible instance is found, the oracle signals that the ciphertext is invalid.
- Otherwise, $\mathcal{S}$ retrieves $\langle \mathsf{id}_B, \lambda_B\rangle$ from L_0, computes the private key $d_B = (g^\zeta)^{\lambda_B}$, uses it to decrypt the ciphertext using $\mathsf{Decrypt}$ in the regular way, and verifies the resulting signed plaintext with Verify. If all is well, the output from $\mathsf{Decrypt}$ is returned; otherwise, it is signaled that the ciphertext is invalid.

• Key extraction queries: suppose $\mathcal{S}$ is queried on an identity id.
  - If id is a guessed identity id_ξ, then $\mathcal{S}$ signals a violation of the assumptions.
  - Otherwise, as L_0 necessarily contains a unique matching entry $\langle \mathsf{id}, \lambda\rangle$, the simulator $\mathcal{S}$ retrieves λ from the list, and computes the corresponding private key as $d = (g^\zeta)^\lambda$, which is then returned.

Analysis:

It is readily observed that the simulation of: (1) the signature/encryption oracle, (2) the key extraction oracle, and (3) the decryption oracle in the case where id_B is not a guessed identity, are all true to life, as long as the adversary making the queries abides by the stated constraints. In particular, the correctness of the signature/encryption oracle when id_A—but not id_B—is a guessed identity, follows from Lemma 16.

The simulation of the decryption oracle in the case where id_B is a guessed identity, is also accurate, but this requires proof. We consider all the anomalies that may cause an adversary to detect a fault in the simulation, and argue against each of them in sequence. The possible anomalies are, *a priori*:

1. a refusal by the oracle to decrypt a valid ciphertext from a non-guessed sender identity id_A addressed to some guessed recipient identity id_B (recall that the case where both id_A and id_B are guessed is unspecified);
2. an incorrect decryption or non-rejection by the oracle of a ciphertext for some guessed recipient id_B (regardless of the sender, the intended recipient or even the validity of the ciphertext for any recipient).

We show that in order for any of these anomalies to occur and be detected by the adversary, the adversary would have to defeat the random oracles, either by correctly guessing an unseen preimage or by finding a collision in one of the hash functions.

1. Regarding the refusal by the oracle to decrypt a valid ciphertext from a non-guessed sender id_A for some guessed recipient $\mathsf{id}_B = \mathsf{id}_\xi$, we show to a contradition that any valid such a ciphertext that an adversary may present must necessarily be recognized as such by the oracle. The argument goes as follows:
   - Since for each triple $\langle \mathsf{id}_A, m, j\rangle$, there exists exactly one value of v such that $\mathsf{Decrypt}[\mathsf{id}_A, m, j, v] = \top$, which furthermore depends on the random oracle $H_1[j, m]$, it is not feasible for an adversary to produce a valid signature for any m, or a ciphertext that decrypts to a validly signed m, without $\langle j, m, v\rangle$ belonging to L_1.
   - Similarly, it is not feasible for the adversary to produce a string z such that $h_4 \oplus z = \langle \mathsf{id}_A, m\rangle$, for $\langle \mathsf{id}_A, m\rangle$ fixed as above, unless the adversary has knowledge of h_4 (or at least a plaintext/ciphertext pair under h_4); for this to happen, the adversary would have to have obtained h_4 from a query to H_4, causing h_4 to appear in some list entry $\langle v, h_4\rangle$ on L_4.

- In turn, the adversary cannot produce a string y such that $h_2 \oplus y = v$ for the specific value of v as above, without the adversary having knowledge of the one-time pad h_2; thus there must be some entry $\langle w, h_2 \rangle$ on L_2.
- For analogous reasons, there must also be an entry $\langle u, h_3 \rangle$ in L_3, such that $u = \mathbf{e}[d_A, i_B] = \mathbf{e}[d_A, g^{\xi}]$. Note that the value of u is easily computable by the simulator as long as id_A and id_B are not both guessed identities.
- In view of the above, it follows that the adversary cannot produce a ciphertext $\langle x, y, z \rangle$ that decrypts to a valid plaintext signed by a non-guessed identity id_A, unless all the various values needed to reconstruct that plaintext figure on the random oracle lists. Therefore, the decryption oracle will always be able to decrypt any valid ciphertext signed by a non-guessed identity, that an adversary may feasibly produce without defeating the random oracles.

In other words, the first type of anomaly—*i.e.*, a refusal by the oracle to decrypt a valid ciphertext from a non-guessed sender id_A—cannot occur with a non-negligible probability in the random oracle model.

2. Regarding the possibility that the oracle returns an incorrect decryption (including a decryption when a rejection is warranted), we show to a contradiction that, for any guessed identity id_ξ, any candidate plaintext $\langle \mathsf{id}_A, m, j, v \rangle \neq \perp$ that may be returned by the simulator, is the unique correct decryption of the query ciphertext for the given recipient $\mathsf{id}_B = \mathsf{id}_\xi$.

- First, we note that for any fixed ciphertext and recipient, there corresponds at most one correct plaintext $\langle \mathsf{id}_A, m, j, v \rangle$, since decryption is deterministic. Thus, there can be no more than one correct entry in the set of candidate plaintexts entertained by the decryption oracle.
- It is also noted that any candidate plaintext $\langle \mathsf{id}_A, m, j, v \rangle$ entertained by the simulator will bear a sender identity id_A that is not a guessed identity.
- Next, we observe that the set of constraints that any candidate plaintext $\langle \mathsf{id}_A, m, j, v \rangle$ must satisfy, when id_A is not guessed, corresponds to a proof that the plaintext is consistent with the output of $\mathsf{Decrypt}$ on the given ciphertext, and is furthermore valid according to Verify. (More precisely, there is a direct correspondence between the constraints enforced by the oracle on the one hand, and the computations performed by $\mathsf{Decrypt}$ and Verify on the other hand.)
- It follows that any (non-reject) plaintext $\langle \mathsf{id}_A, m, j, v \rangle$ returned by the decryption oracle is necessarily the correct and valid decryption of the ciphertext for the given recipient id_B, where id_A is not a guessed identity. By unicity, it is also the case that the decryption procedure in $\mathcal{S}$ never has to break ties; therefore the output is well-defined.

In other words, the second type of anomaly—*i.e.*, an incorrect output that is not a rejection—cannot occur with a non-negligible probability in the random oracle model.

In summary, as long as the queries remain within the given specifications, the simulation provided by $\mathcal{S}$ is indistinguishable from the reality, barring the improbable occurence of a random oracle anomaly.

□

B.1 Confidentiality

Proof of Theorem 11. We outline the construction of an algorithm $\mathcal{B}$, that, given an instance $\langle g, g^\alpha, g^\beta, g^\gamma \rangle$, probabilistically computes $\mathbf{e}[g, g]^{\alpha\,\beta\,\gamma}$, by posing as a challenger for $\mathcal{A}$ which it uses as a subroutine while emulating the required oracles. (It is assumed that $\mathcal{B}$ is given the other common parameters: p, $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbf{e}$.)

CONSTRUCTION:

The construction of $\mathcal{B}$ is based on the generic simulator $\mathcal{S}$ described in Lemma 17, augmented as follows.

First, $\mathcal{B}$ sets up a simulator $\mathcal{S}$ as in Lemma 17, setting its formal public parameters $\langle g, g^\zeta \rangle$ to $\langle g, g^\gamma \rangle$ and its crucial set $\{g^{\xi_i}\}$ to $\{g^\gamma\}$. In addition, $\mathcal{B}$ picks a number η_β uniformly at random in $\{1, ..., \mu_0\}$. Then, $\mathcal{B}$ runs $\mathcal{A}$ on the IBSE public parameters $\langle g, g^\gamma \rangle$, servicing all queries from $\mathcal{A}$ exactly as $\mathcal{S}$ in Lemma 17, except for the following modifications. As to queries to the random oracles:

- Queries to H_0:
  - The η_β-th distinct query to H_0 is automatically evaluated to the crucial element g^β, as permitted per Lemma 17. The corresponding queried identity is denoted id_β and is singled out as the 'guessed' recipient.

As for the IBSE oracles, the modifications are:

- Key extraction queries:
  - If a key extraction query is made on the guessed identity id_β, then $\mathcal{B}$ terminates its interaction with $\mathcal{A}$, having failed to guess the targeted recipient among those in L_0.

At some point, $\mathcal{A}$ produces two identities id_A and id_B and two equal-length messages m_0 and m_1 on which it wishes to be challenged. $\mathcal{B}$ responds with the challenge ciphertext $\langle (g^\alpha), y, z\rangle$, where y and z are random strings of appropriate size.

All further queries by $\mathcal{A}$ (subject to the additional restrictions stated in the security model) are processed as previously described.

Finally, $\mathcal{A}$ returns its final guess. $\mathcal{B}$ ignores the answer from $\mathcal{A}$, picks an entry $\langle \bar{w}, h_2\rangle$ uniformly at random in L_2, and returns $\bar{w}$ as its guess for the solution to the given BDH instance.

ANALYSIS:

Per Lemma 17, whenever the recipient identity id_B selected by $\mathcal{A}$ is the one id_β guessed by $\mathcal{B}$, the simulation provided by $\mathcal{B}$ is indistinguishable from a genuine attack scenario, except for the challenge ciphertext eventually presented to $\mathcal{A}$. (Indeed, per the irreflexivity assumption, sender and recipient identities are always different for a given ciphertext, so that all the conditions of Lemma 17 are met.)

Since the challenge ciphertext presented to $\mathcal{A}$ is randomly distributed in the space of ciphertexts of the correct size, $\mathcal{A}$ cannot gain any advantage in this simulation. Thus, any adversary that has advantage ϵ in the real IND-IBSE-CCA game must necessarily recognize with probability at least ϵ that the challenge ciphertext provided by $\mathcal{B}$ is incorrect.

Recognizing that the challenge ciphertext of the form $\langle x, y, z\rangle$ with $x = g^\alpha$ is incorrect requires a random oracle query $H_2[\bar{w}]$ with $\bar{w} = \mathbf{e}[x, d_\beta] = \mathbf{e}[g, g]^{\alpha\,\beta\,\gamma}$. Any such query by $\mathcal{A}$ leaves an entry $\langle \bar{w}, h_2\rangle$ on L_2, from which $\mathcal{B}$ can then extract $\mathbf{e}[g, g]^{\alpha\,\beta\,\gamma} = \bar{w}$ with (conditional) probability $1/\mu_2$.

Taking into account the marginal probability $1/\mu_0$ of the conditioning event that $\mathcal{B}$ makes the correct choice for the guessed identity id_β, the probability of $\mathcal{B}$ correctly solving the BDH instance becomes:

$$\mathbf{Adv}[\mathcal{B}] = \frac{1}{\mu_0\,\mu_2}\,\epsilon\ ,$$

where $\epsilon = \mathbf{Adv}[\mathcal{A}]$ is the advantage of $\mathcal{A}$ in the real IND-IBSE-CCA game.

□

B.2 Non-Repudiation

Proof of Theorem 12. We outline the construction of $\mathcal{B}$, that, given a BDH instance $\langle g, g^\alpha, g^\beta, g^\gamma\rangle$, probabilistically computes $\mathbf{e}[g, g]^{\alpha\,\beta\,\gamma}$, posing as a challenger for $\mathcal{A}$ which it uses as a subroutine. In fact, $\mathcal{B}$ computes $g^{\alpha\,\gamma}$, thus solving the computational Diffie-Hellman problem in $\mathbb{G}_1$, which is clearly at least as hard as computing $\mathbf{e}[g, g]^{\alpha\,\beta\,\gamma}$.

The reduction is similar to the proof of unforgeability in [8], itself based on the powerful "Forking Lemma" by Pointcheval and Stern [26].

PRELIMINARIES:

The forking lemma essentially says the following [26, §3.2.1]. Consider a signature scheme producing signatures of the form $\langle m, \sigma_1, h, \sigma_2\rangle$, where each of σ_1, h, σ_2 corresponds to one of the three phases of some honest-verifier zero-knowledge identification protocol—*i.e.*, σ_1 is a commitment by the prover/signer,

$h = H[m, \sigma_1]$ serves to simulate a random challenge by the verifier, and σ_2 is the prover/signer's response to the challenge. Suppose that $\mathcal{A}$ is an adaptive CMA existential forger, that makes μ_S signature queries and μ_R random oracle queries, and forges a signature $\langle m, \sigma_1, h, \sigma_2 \rangle$ in time τ with probability $\epsilon \geq 10\,(\mu_S + 1)\,(\mu_S + \mu_R)/2^n$. If the triples $\langle \sigma_1, h, \sigma_2 \rangle$ can be perfectly simulated without knowing the private key (*e.g.*, by manipulating the random oracles instead), then there exists an algorithm $\mathcal{A}'$ that, using $\mathcal{A}$ as a subroutine, produces two valid signatures $\langle m, \sigma_1, h, \sigma_2 \rangle$ and $\langle m, \sigma_1, h', \sigma_2' \rangle$ such that $h \neq h'$, in expected time $\tau' \leq 120686\,\mu_R\,\tau/\epsilon$.

First, we observe that the IBSE Sign algorithm produces signatures of the form $\langle m, \sigma_1, h, \sigma_2 \rangle$, which corresponds to the required three-phase honest-verifier zero-knowledge identification protocol, $\sigma_1 = j$ being the prover's commitment, $h = H_1[j, m]$ a hash value substituted for the verfier's challenge, and $\sigma_2 = v$ the prover's response.

The rest of the proof then consists of the following steps:

1. A simulation step, in which we show how to simulate the triples $\langle \sigma_1, h, \sigma_2 \rangle$ without the secret key of the sender (and thus, also without the master secret). By the forking lemma, this gives us a machine $\mathcal{A}'$ that produces two valid signatures $\langle m, \sigma_1, h, \sigma_2 \rangle$ and $\langle m, \sigma_1, h', \sigma_2' \rangle$ with $h \neq h'$, as described above.
2. A reduction step, in which we show how to solve the BDH problem by interacting with $\mathcal{A}'$.

SIMULATION:

We start by describing a machine $\mathcal{B}'$ that provides a faithful simulation to the forger $\mathcal{A}$ (where $\mathcal{A}$ is assumed, for simplicity, to always run a decryption query on its forged ciphertext before returning it to $\mathcal{B}'$). The construction is based on the simulator $\mathcal{S}$ described in Lemma 17, extended as follows.

First, $\mathcal{B}'$ sets up a simulator $\mathcal{S}$ as in Lemma 17; $\mathcal{S}$ is supplied with the public parameters $\langle g, g^\gamma \rangle$ and a crucial singleton $\{g^\alpha\}$. In addition, $\mathcal{B}'$ picks a number η_α uniformly at random in $\{1, ..., \mu_0\}$. $\mathcal{B}'$ then runs $\mathcal{A}$ on the IBSE public parameters $\langle g, g^\gamma \rangle$, servicing all queries from $\mathcal{A}$ exactly as $\mathcal{S}$ in Lemma 17, except for the following modifications:

- Queries to H_0:
  - If the query is the η_α-th (distinct) such query so far, it is evaluated as g^α, as per Lemma 17. The corresponding identity is denoted id_α and is called the 'guessed' sender.
- Key extraction queries:
  - If a key extraction query concerns the guessed identity id_α, then $\mathcal{B}'$ terminates its interaction with $\mathcal{A}$, having failed to guess the targeted sender among those in L_0.

Eventually, $\mathcal{A}$ returns a forgery, consisting of a ciphertext c and a recipient identity id_B. $\mathcal{B}'$ decrypts the ciphertext for id_B (by invoking its own decryption oracle, supplied by the simulator $\mathcal{S}$), which causes the plaintext forgery $\langle \mathsf{id}_A, m, j, v \rangle$ to be revealed. Note that if $\mathcal{B}'$ has made the correct guess, *i.e.*, $\mathsf{id}_A = \mathsf{id}_\alpha$, then necessarily $\mathsf{id}_B \neq \mathsf{id}_\alpha$ and the decryption works.

It is easy to see that the simulation provided by $\mathcal{B}'$ is true to life; in particular, the distribution of signatures in the case $\mathsf{id}_A = \mathsf{id}_\alpha$ is the same as in reality.

REDUCTION:

From the results of Lemma 17, it is easy to see that the simulation provided by $\mathcal{B}'$ is true to life within the constraints of the EUF-IBSE-CMA attack, provided that $\mathcal{B}'$ correctly guessed the targeted sender identity. It follows from the forking lemma that if $\mathcal{A}$ is a sufficiently efficient forger in the above interaction, then we can construct a Las Vegas machine $\mathcal{A}'$ that outputs two signed messages $\langle \langle \mathsf{id}_A, m \rangle, j, h, v \rangle$ and $\langle \langle \mathsf{id}_A, m \rangle, j, h', v' \rangle$ with $h \neq h'$.

It is remarked here that we are implicitly coalescing the signing identity id_A and the message proper m into a 'generalized' forged message $\langle \mathsf{id}_A, m \rangle$, for the purpose of applying the forking lemma. This is in order to hide the identity-based aspect of the EUF-IBSE-CMA attack, and simulate the setting of an (identity-less) adaptive-CMA existential forgery for which the forking lemma is proven [26, §3.2.1].

Thus, given $\mathcal{A}$, we derive a machine $\mathcal{A}'$, and use it to construct a second machine $\mathcal{B}$ that is our reduction from the BDH problem. $\mathcal{B}$ proceeds as follows.

1. $\mathcal{B}$ runs $\mathcal{A}'$ to obtain two distinct forgeries $\langle\langle \mathsf{id}_A, m\rangle, j, h, v\rangle$ and $\langle\langle \mathsf{id}_A, m\rangle, j, h', v'\rangle$.
2. $\mathcal{B}$ derives the value of $g^{\alpha\,\gamma}$ as $(v'\,v^{-1})^{(h'-h)^{-1}}$.

ANALYSIS:

First, we note that if $\mathcal{A}'$ returns two valid forged signatures with $h \neq h'$, then $\mathcal{B}$ successfully computes $g^{\alpha\,\gamma}$, and thus $\mathbf{e}[g, g]^{\alpha\,\beta\,\gamma}$. Indeed, for some $r \in \mathbb{F}_p^\star$:

$$(v'\,v^{-1})^{(h'-h)^{-1}} = ((g^{\alpha\,\gamma})^{h'+r-h-r})^{(h'-h)^{-1}} \quad = g^{\alpha\,\gamma} \ .$$

It remains to show that the simulation provided to $\mathcal{A}$ by $\mathcal{B}'$ in the first phase of the proof succeeds with non-negligible probability. To this end, we note that for the simulation to succeed, it suffices that $\mathcal{A}$ ask no key extraction query on id_α. Since this holds with probability at least $1/\mu_0$, as there are at most μ_0 entries in L_0, the expected number of attempts required to achieve a successful simulation is at most μ_0.

Incorporating the bound from the forking lemma, we obtain that, if $\mathcal{A}$ succeeds in time $\leq \tau$ with probability $\geq \epsilon = 10\,(\mu_{\text{se}} + 1)\,(\mu_{\text{se}} + \mu_1)/2^n$, then $\mathcal{B}$ solves the bilinear Diffie-Hellman problem in expected time $\leq 120686\,\mu_0\,\mu_1\,\tau/\epsilon$.

□

B.3 Unlinkability

Proof of Theorem 13. This theorem is a direct consequence of the existence of an efficient $\mathsf{EncryptToSelf}$ function, as shown in Lemma 16.

The $\mathsf{EncryptToSelf}$ function allows any recipient id_B to transform any plaintext signed by id_A into a valid ciphertext from id_A addressed to id_B. By Lemma 16, the ciphertext thus obtained is easily verified to be identical to the ciphertext that id_A would have produced from the same message with the same signature (recall that $\mathsf{Encrypt}$ and $\mathsf{EncryptToSelf}$ are deterministic).

□

B.4 Authentication

Proof of Theorem 14. We outline the construction of $\mathcal{B}$, that, given a BDH instance $\langle g, g^\alpha, g^\beta, g^\gamma\rangle$, probabilistically computes $\mathbf{e}[g, g]^{\alpha\,\beta\,\gamma}$, posing as a challenger for $\mathcal{A}$ which it uses as a subroutine.

CONSTRUCTION:

For simplicity, we assume that the adversary $\mathcal{A}$ always runs a decryption query on its forged ciphertext before returning it to $\mathcal{B}$.

The construction of $\mathcal{B}$ is based on the generic simulator described in Lemma 17, augmented as follows.

First, $\mathcal{B}$ sets up a simulator $\mathcal{S}$ as in Lemma 17, which is given the public parameters $\langle g, g^\gamma\rangle$ and the crucial set $\{g^\alpha, g^\beta\}$. In addition, $\mathcal{B}$ picks two numbers η_α and η_β uniformly at random in $\{1, ..., \mu_0\}$. It also initializes an empty special list $L_\star$ for its own use.

$\mathcal{B}$ then runs $\mathcal{A}$ on the IBSE public parameters $\langle g, g^\gamma\rangle$, servicing all queries from $\mathcal{A}$ exactly as $\mathcal{S}$ in Lemma 17, except for the following modifications. Regarding queries to the random oracles:

- Queries to H_0:
  - If the query is either the η_α-th or η_β-th such (distinct) query encountered so far, then the query is evaluated it as g^α or g^β, respectively, as per Lemma 17. The corresponding query identities are denoted id_α and id_β, and called the 'guessed' sender and recipient, respectively.

As for the IBSE oracles, the modifications are:

- Key extraction queries:
  - If a key extraction query identity is made on either id_α or id_β, then $\mathcal{B}$ terminates its interaction with $\mathcal{A}$, having failed to guess the targeted sender or recipient among those in L_0.

- Signature/encryption queries:
  - If the specified sender identity is id_α: $\mathcal{B}$ proceeds as $\mathcal{S}$ would, per Lemma 17, which results in a valid signature $\langle j, v\rangle$ for the query message m and signer identity id_α (and causes the addition of a new entry $\langle j, v, h_1\rangle \in L_1$). For future use, recall that two random numbers $r, h \in \mathbb{F}_p^\star$ are created in this process (*cf.* Lemma 17). Then:
    * If the specified recipient identity is id_β, $\mathcal{B}$ proceeds with the encryption task as follows: $\mathcal{B}$ successively picks two random elements $u \in \mathbb{G}_2^\star$ and $w \in \mathbb{G}_2^\star$, obtains $h_2 = H_2[w]$, $h_3 = H_3[u]$, $h_4 = H_4[v]$, and computes $x = j^{h_3}$, $y = h_2 \oplus v$, $z = h_4 \oplus \langle \mathsf{id}_\alpha, m\rangle$. Finally, $\mathcal{B}$ adds the tuple $\langle \mathsf{id}_\alpha, m, j, v, u, w, r, h\rangle$ to the special list $L_\star$, and returns the (fake) ciphertext $\langle x, y, z\rangle$.
    * Otherwise, $\mathcal{B}$ continues the encryption process exactly as $\mathcal{S}$ would, *i.e.*, using $\mathsf{EncryptToSelf}$ to produce a (correct) ciphertext from the signature obtained thus far. In addition, $\mathcal{B}$ adds the tuple $\langle \mathsf{id}_\alpha, m, j, v, u, w, r, h\rangle$ to the special list $L_\star$, where u and w are computed during the execution of $\mathsf{EncryptToSelf}$ (*cf.* Lemma 16).

    Note that a tuple $\langle \mathsf{id}_\alpha, m, j, v, u, w, r, h\rangle$ is added to the special list $L_\star$ in all cases where the sender identity is the guessed sender id_α.
  - If the specified sender identity is id_β: $\mathcal{B}$ proceeds as above, where the roles of id_α and id_β are interchanged. (In this case a tuple $\langle \mathsf{id}_\beta, m, j, v, u, w, r, h\rangle$ is added to the special list $L_\star$.)
- Decryption queries:
  - If the recipient identity specified to the oracle is id_β, $\mathcal{B}$ proceeds as follows. First, the lists are searched for all instances of $\langle \mathsf{id}_A, m, j, v\rangle$ such that $\langle j, m, h_1\rangle \in L_1$, $\langle w, h_2\rangle \in L_2$, $\langle u, h_3\rangle \in L_3$, $\langle v, h_4\rangle \in L_4$, for some h_1, h_2, h_3, h_4, u, w, under the constraints that $h_2 \oplus y = v$, $x^{h_3^{-1}} = j$, $h_4 \oplus z = \langle \mathsf{id}_A, m\rangle$, and $\mathsf{Verify}[\mathsf{id}_A, m, j, v] = \top$. Then, for each such $\langle \mathsf{id}_A, m, j, v\rangle$:
    * If $\mathsf{id}_A = \mathsf{id}_\beta$, then the instance $\langle \mathsf{id}_A, m, j, v\rangle$ is rejected from further consideration as a decryption candidate.
    * If $\mathsf{id}_A = \mathsf{id}_\alpha$, then the instance $\langle \mathsf{id}_A, m, j, v\rangle$ is retained as a decryption candidate if $L_\star$ contains a tuple $\langle \mathsf{id}_\alpha, m, j, v, u, w, r, h\rangle$, for some r, h.
    * Otherwise, the instance $\langle \mathsf{id}_A, m, j, v\rangle$ is retained as a decryption candidate if $\mathbf{e}[d_A, g^\beta] = u$ and $\mathbf{e}[v\,(d_A)^{-h_1}, g^\beta]^{h_3} = w$, where $d_A = (g^\gamma)^{\lambda_A}$ is the private key of id_A, with λ_A found in an entry $\langle \mathsf{id}_A, \lambda_A\rangle \in L_0$.

    Finally, the oracle selects a decryption candidate $\langle \mathsf{id}_A, m, j, v\rangle$ that satisfies all the above conditions (giving priority to any decryption with $\mathsf{id}_A \neq \mathsf{id}_\alpha$, then breaking ties arbitrarily), and returns it as the decrypted signed plaintext for the query. In case no eligible candidate is found, the oracle signals that the ciphertext is invalid.
  - If the recipient identity is id_α: $\mathcal{B}$ proceeds as above, where the roles of id_α and id_β are interchanged wherever they appear.

(Recall that any legitimate query that does not fit any of the special cases above is treated by default as described in Lemma 17.)

At some point, $\mathcal{A}$ returns its forgery, consisting of a recipient identity id_B and a ciphertext $c = \langle x, y, z\rangle$, where it is assumed that $\mathcal{A}$ has previously made a decryption query on c for recipient id_B.

$\mathcal{B}$ then produces its own answer in the following randomized manner:

Choice 1 With probability $\mu_3/(\mu_1\,\mu_2 + \mu_3)$, $\mathcal{B}$ picks an entry $\langle \bar{u}, h_3\rangle$ uniformly at random in L_3, and returns the value $\bar{u}$.

Choice 2 With probability $\mu_1\,\mu_2/(\mu_1\,\mu_2 + \mu_3)$, $\mathcal{B}$ picks an entry $\langle \bar{w}, h_2\rangle$ uniformly at random in L_2, and an entry $\langle \mathsf{id}, m, j, v, u, w, r, h\rangle$ uniformly at random in $L_\star$ such that $\mathsf{id} = \mathsf{id}_\alpha$ and $\exists h_3 : \langle u, h_3\rangle \in L_3$, and returns the value $\bar{w}^{-(h\,h_3)^{-1}}\,\mathbf{e}[g^\alpha, g^\gamma]^{h^{-1}\,r}$.

Analysis:

We assume that $\mathcal{B}$ correctly guessed id_α and id_β for the forged sender and recipient identities respectively, which it does with probability $1/(\mu_0(\mu_0 - 1))$.

First, we observe that the simulation provided by $\mathcal{B}$ is indistinguishable from reality for all queries that do not involve id_β, since then $\mathcal{B}$ merely replicates the behavior of $\mathcal{S}$ from Lemma 17 in those cases. Detectable discrepancies with a true attack scenario only occur in the following situations:

1. Signature/encryption queries when sender and recipient identities are id_α and id_β (in either order): in this case the oracle is unable to construct a correct ciphertext, and returns a fake one instead.
2. Decryption queries for recipient id_β (resp. id_α), where the plaintext candidates entertained by the decryption oracle all bear signatures by id_α (resp. id_β): in this case the oracle always rejects the ciphertext, unless it can positively trace it (via the random oracle simulation lists L_1, L_2, L_3, L_4, and the special list $L_\star$) to a prior signature/encryption query.

Notice that the verification against $L_\star$ ensures that there is no risk of decrypting an incorrect ciphertext for recipient id_β or id_α that did not result from a previous signature/encryption query.

We now show that in all cases where $\mathcal{A}$ either manages a successful forgery or detects any of the above discrepancies, $\mathcal{B}$ is in a position to solve the BDH instance with non-negligible probability. The cases to consider are:

1. $\mathcal{A}$ detects that a ciphertext from id_α to id_β is correct even though it is rejected by the decryption oracle: for this to happen the adversary must learn the value of $H_3[\bar{u}]$ with $\bar{u} = \mathbf{e}[i_\alpha, d_\beta] = \mathbf{e}[g,g]^{\alpha\,\beta\,\gamma}$. Any such query by $\mathcal{A}$ leaves an entry $\langle \bar{u}, h_3\rangle$ on L_3, from which $\mathcal{B}$ can extract $\mathbf{e}[g,g]^{\alpha\,\beta\,\gamma} = \bar{w}$ with (conditional) probability $1/\mu_3$ when $\mathcal{B}$ executes Choice 1.
2. $\mathcal{A}$ detects that a ciphertext $\langle x, y, z\rangle$ returned by a signature/encryption query with sender identity id_α and recipient identity id_β is incorrect: for this to happen, the adversary has to query the oracle for $H_2[\bar{w}]$ with $\bar{w} = \mathbf{e}[x, g^{\beta\,\gamma}]$. Since in this case $x = (g^r\,(g^\alpha)^{-h})^{h_3}$ for some $\langle \mathsf{id}_\alpha, m, j, v, u, w, r, h\rangle \in L_\star$ with $\langle u, h_3\rangle \in L_3$, it follows that $\bar{w} = \mathbf{e}[x, d_\beta] = \mathbf{e}[g^r, g^{\beta\,\gamma}]^{h_3}\,\mathbf{e}[g^{-\alpha\,h}, g^{\beta\,\gamma}]^{h_3}$. Thus, any such query leaves an entry $\langle \bar{w}, h_2\rangle$ on L_2, from which $\mathcal{B}$ can extract
$$\mathbf{e}[g,g]^{\alpha\,\beta\,\gamma} = \bar{w}^{-(h\,h_3)^{-1}}\,\mathbf{e}[g^\beta, g^\gamma]^{h^{-1}\,r}\ ,$$
with (conditional) probability $1/(\mu_2\,|L_\star|) \geq 1/(\mu_1\,\mu_2)$, since the size of the list $L_\star$ is never greater than that of L_1. This happens conditionally upon $\mathcal{B}$ executing Choice 2.
3. $\mathcal{A}$ successfully forges a ciphertext that would decrypt under id_β to a validly signed plaintext by id_α: since $\mathcal{A}$ is assumed to run the decryption oracle on any forgery it returns, this case reduces to the detection of an incorrect rejection, as above, and is thus covered by $\mathcal{B}$ executing Choice 2.

The cases where id_β is the sender and id_α the recipient are analogous.

Accounting for the probability that $\mathcal{B}$ correctly guesses id_α and id_β, it follows from the above analysis that $\mathcal{B}$ solves the BDH problem with probability
$$\begin{aligned}\mathbf{Adv}[\mathcal{B}] &= \frac{1}{\mu_0\,(\mu_0-1)}\left(\frac{\mu_3}{\mu_1\,\mu_2+\mu_3}\,\frac{1}{\mu_3} + \frac{\mu_1\,\mu_2}{\mu_1\,\mu_2+\mu_3}\,\frac{1}{\mu_1\,\mu_2}\right)\epsilon \\ &= \frac{2\,\epsilon}{\mu_0\,(\mu_0-1)\,(\mu_1\,\mu_2+\mu_3)}\ ,\end{aligned}$$
where $\epsilon = \mathbf{Adv}[\mathcal{A}]$ is the advantage of $\mathcal{A}$ in a real AUTH-IBSE-CMA attack.

□

B.5 Anonymity

Proof of Theorem 15. We outline the construction of $\mathcal{B}$, that, given a BDH instance $\langle g, g^\alpha, g^\beta, g^\gamma\rangle$, probabilistically computes $\mathbf{e}[g,g]^{\alpha\,\beta\,\gamma}$, by using $\mathcal{A}$ as a subroutine.

CONSTRUCTION:

First, $\mathcal{B}$ sets up a simulator $\mathcal{S}$ as in Lemma 17, which is given the public parameters $\langle g, g^\gamma\rangle$ and the crucial set $\{g^\alpha, g^\beta\}$. In addition, $\mathcal{B}$ picks two distinct numbers η_α and η_β uniformly at random in $\{1, ..., \mu_0\}$.

It also initializes a special list $L_\star$ for its own use. $\mathcal{B}$ then runs $\mathcal{A}$ on the IBSE public parameters $\langle g, g^\gamma \rangle$, servicing all queries from $\mathcal{A}$ exactly as $\mathcal{S}$ in Lemma 17, except for the following modifications. For queries to the random oracles:

- Queries to H_0:
  - On the η_α-th distinct query to H_0, $\mathcal{B}$ directs $\mathcal{S}$ to evaluate it as g^α as in Lemma 17. The query identity is denoted id_α and is called the first 'guessed' recipient.
  - On the η_β-th distinct query to H_0, $\mathcal{B}$ similarly directs $\mathcal{S}$ to evaluate it as g^β. The query identity is denoted id_β and is called the second 'guessed' recipient.

The IBSE oracles are modified from $\mathcal{S}$ as follows:

- Key extraction queries:
  - If a key extraction query is made on either id_α or id_β, then $\mathcal{B}$ terminates its interaction with $\mathcal{A}$, having failed to guess the two targeted recipients among those in L_0.
- Signature/encryption queries:
  - If the designated sender and recipient identities are id_α and id_β (in either order; but it is assumed for the sake of exposition that id_α is the sender and id_β the recipient): $\mathcal{B}$ proceeds as $\mathcal{S}$ would, per Lemma 17, toward the production a valid signature $\langle j, v \rangle$ for the query message m and signer identity id_α. (We note that this causes the addition of a new entry $\langle j, v, h_1 \rangle \in L_1$, and involves the selection of two random numbers denoted $r, h \in \mathbb{F}_p^\star$, *cf.* Lemma 17.) $\mathcal{B}$ then continues as follows. $\mathcal{B}$ successively picks two random elements $u \in \mathbb{G}_2^\star$ and $w \in \mathbb{G}_2^\star$, obtains $h_2 = H_2[w]$, $h_3 = H_3[u]$, $h_4 = H_4[v]$, and computes $x = j^{h_3}$, $y = h_2 \oplus v$, $z = h_4 \oplus \langle \mathsf{id}_\alpha, m \rangle$. Finally, $\mathcal{B}$ adds the tuple $\langle \mathsf{id}_\alpha, m, j, v, u, w, r, h \rangle$ to the special list $L_\star$, and returns the (fake) ciphertext $\langle x, y, z \rangle$.
- Decryption queries:
  - If the requested recipient identity is either id_α or id_β (assuming for the sake of exposition that it is id_β): First, $\mathcal{B}$ determines all instances of $\langle \mathsf{id}_A, m, j, v \rangle$ such that $\langle j, m, h_1 \rangle \in L_1$, $\langle w, h_2 \rangle \in L_2$, $\langle u, h_3 \rangle \in L_3$, $\langle v, h_4 \rangle \in L_4$, for some h_1, h_2, h_3, h_4, u, w, under the constraints that $h_2 \oplus y = v$, $x^{h_3^{-1}} = j$, $h_4 \oplus z = \langle \mathsf{id}_A, m \rangle$, and $\mathsf{Verify}[\mathsf{id}_A, m, j, v] = \top$. Then, for each such $\langle \mathsf{id}_A, m, j, v \rangle$:
    * If $\mathsf{id}_A = \mathsf{id}_\beta$, $\langle \mathsf{id}_A, m, j, v \rangle$ is rejected from further consideration as a decryption candidate.
    * If $\mathsf{id}_A = \mathsf{id}_\alpha$, $\langle \mathsf{id}_A, m, j, v \rangle$ is retained as a decryption candidate only if there exists a tuple $\langle \mathsf{id}_\alpha, m, j, v, u, w, r, h \rangle \in L_\star$, for some r and h.
    * Otherwise, $\langle \mathsf{id}_A, m, j, v \rangle$ is retained provided that $\mathbf{e}[v\,(d_A)^{-h_1}, g^\beta]^{h_3} = w$ and $\mathbf{e}[d_A, g^\beta] = u$, where $d_A = (g^\gamma)^{\lambda_A}$ is the private key of id_A, with λ_A found in some entry $\langle \mathsf{id}_A, \lambda_A \rangle \in L_0$.

    Finally, the oracle selects any decryption candidate $\langle \mathsf{id}_A, m, j, v \rangle$ satisfying all of the applicable conditions above (giving priority to any decrypted plaintext such that $\mathsf{id}_A \neq \mathsf{id}_\alpha$, then breaking ties arbitrarily), and returns the selected candidate as the answer to the decryption query. In case no eligible candidate is found, the oracle signals that the ciphertext is invalid.

(As before, recall that any legitimate query that does not fit any of the above cases is processed by default as described in Lemma 17.)

At some point, $\mathcal{A}$ will produce a message m, two sender identities id_{A_0} and id_{A_1}, and two recipient identities id_{B_0} and id_{B_1} on which it wishes to be challenged. If $\{\mathsf{id}_{B_0}, \mathsf{id}_{B_1}\} \neq \{\mathsf{id}_\alpha, \mathsf{id}_\beta\}$, then $\mathcal{B}$ declares failure. Otherwise, $\mathcal{B}$ responds with a challenge ciphertext $\langle (g^\alpha)^{\bar{r}}, y, z \rangle$, where y and z are random strings of appropriate size, and $\bar{r} \in \mathbb{F}_p^\star$ is a random value saved for future use.

All further queries by $\mathcal{A}$ (subject to the appropriate additional restrictions) are processed as previously described.

Finally, $\mathcal{A}$ returns its guess. $\mathcal{B}$ ignores it, and produces its own answer in the following randomized manner:

Choice 1 With probability $\mu_3/(\mu_1\,\mu_2 + 2\,\mu_2 + \mu_3)$, $\mathcal{B}$ picks an entry $\langle \bar{u}, h_3 \rangle$ uniformly at random in L_3, and returns the value $\bar{u}$.

Choice 2 With probability $\mu_1\,\mu_2/(\mu_1\,\mu_2+2\,\mu_2+\mu_3)$, $\mathcal{B}$ picks an entry $\langle\bar{w},h_2\rangle$ uniformly at random in L_2, and an entry $\langle \mathsf{id},m,j,v,u,w,r,h\rangle$ uniformly at random in $L_\star$ such that $\mathsf{id}=\mathsf{id}_\alpha$ and $\exists h_3:\langle u,h_3\rangle\in L_3$, and returns the value $\bar{w}^{-(h\,h_3)^{-1}}\,\mathbf{e}[g^\alpha,g^\gamma]^{h^{-1}\,r}$.

Choice 3 With probability $2\,\mu_2/(\mu_1\,\mu_2+2\,\mu_2+\mu_3)$, $\mathcal{B}$ picks an entry $\langle\bar{w},h_3\rangle$ uniformly at random in L_3, and returns the value $\bar{w}^{\bar{r}^{-1}}$.

Analysis:

We assume that $\mathcal{B}$ has made the correct choices for id_α and id_β, which happens with probability $1/(\mu_0(\mu_0-1))$.

First, we observe that the simulation provided by $\mathcal{B}$ is indistinguishable from reality for all queries that do not simultaneously involve both id_α and id_β; this is because then $\mathcal{B}$ merely replicates the behavior of $\mathcal{S}$ from Lemma 17. Detectable discrepancies from a genuine attack occur only in the following situations:

1. Signature/encryption queries when the sender and recipient identities are id_α and id_β (in either order): in this case $\mathcal{B}$'s oracle is unable to construct a correct ciphertext and returns a fake one instead.
2. Decryption queries for recipient id_β (resp. id_α), where the decryption candidates retained by the decryption oracle all bear signatures from id_α (resp. id_β): in this case the oracle always rejects the ciphertext, unless it can positively trace it (via the random oracle simulation lists and the special list $L_\star$) to a prior signature/encryption query.

Notice that the verification against $L_\star$ ensures that there is no risk of decrypting an incorrect ciphertext for recipient id_β or id_α that did not result from a previous signature/encryption query.

Now, we argue that in all cases where $\mathcal{A}$ is in a position to either make a correct guess other than by random guessing or detect any of the above discrepancy, $\mathcal{B}$ is in a position to solve the BDH instance with non-negligible probability. The cases to consider are:

1. Detection that a ciphertext from id_α to id_β is correct even though it is rejected by the decryption oracle: for this to happen the adversary must learn the value of $H_3[\bar{u}]$ with $\bar{u}=\mathbf{e}[i_\alpha,d_\beta]=\mathbf{e}[g,g]^{\alpha\,\beta\,\gamma}$. Any such query made by $\mathcal{A}$ leaves an entry $\langle\bar{u},h_3\rangle$ on L_3, from which $\mathcal{B}$ can extract $\mathbf{e}[g,g]^{\alpha\,\beta\,\gamma}=\bar{w}$ with probability $1/\mu_3$ conditionally on $\mathcal{B}$ selecting Choice 1.
2. Detection that a ciphertext $\langle x,y,z\rangle$ returned by a signature/encryption queries with sender identity id_α and recipient identity id_β is incorrect: for this to happen, the adversary has to make the oracle query $H_2[\bar{w}]$ with $\bar{w}=\mathbf{e}[x,g^{\beta\,\gamma}]$. Since in this case $x=(g^r\,(g^\alpha)^{-h})^{h_3}$ for some $\langle\mathsf{id}_\alpha,m,j,v,u,w,r,h\rangle\in L_\star$ with $\langle u,h_3\rangle\in L_3$, it follows that $\bar{w}=\mathbf{e}[x,d_\beta]=\mathbf{e}[g^r,g^{\beta\,\gamma}]^{h_3}\,\mathbf{e}[g^{-\alpha\,h},g^{\beta\,\gamma}]^{h_3}$. Thus, any such query made by $\mathcal{A}$ leaves an entry $\langle\bar{w},h_2\rangle$ on L_2, from which $\mathcal{B}$ can extract
$$\mathbf{e}[g,g]^{\alpha\,\beta\,\gamma}=\bar{w}^{-(h\,h_3)^{-1}}\,\mathbf{e}[g^\beta,g^\gamma]^{h^{-1}\,r}\ ,$$
with probability $1/(\mu_2\,|L_\star|)\geq 1/(\mu_1\,\mu_2)$, conditionally on $\mathcal{B}$ selecting Choice 2.
3. Acquisition of any information about the challenge ciphertext $\langle x,y,z\rangle$, allegedly encrypted for either id_α or id_β: for this to happen, the adversary must make at least one of the two oracle queries $H_2[\bar{w}_\alpha]$ and $H_2[\bar{w}_\beta]$, where $\bar{w}_\alpha=\mathbf{e}[x,d_\alpha]$ and $\bar{w}_\beta=\mathbf{e}[x,d_\beta]$. The two are equivalent from $\mathcal{A}$'s viewpoint, but only the latter is useful to $\mathcal{B}$. In that case, since $x=(g^\alpha)^{\bar{r}}$, it follows that $\bar{w}=\mathbf{e}[x,d_\beta]=\mathbf{e}[g,g]^{\bar{r}\,\alpha\,\beta\,\gamma}$. Thus, any such query made by $\mathcal{A}$ leaves an entry $\langle\bar{w},h_2\rangle$ on L_2, from which $\mathcal{B}$ can extract $\mathbf{e}[g,g]^{\alpha\,\beta\,\gamma}=\bar{w}^{\bar{r}^{-1}}$ with probability $1/(2\,\mu_2)$, conditionally on $\mathcal{B}$ selecting Choice 3.

Summing up over the three events, it follows from the above analysis that $\mathcal{B}$ solves the BDH problem with probability
$$\mathbf{Adv}[\mathcal{B}]=\frac{3\,\epsilon}{\mu_0\,(\mu_0-1)\,(\mu_1\,\mu_2+2\,\mu_2+\mu_3)}\ ,$$
where $\epsilon=\mathbf{Adv}[\mathcal{A}]$ is the advantage of $\mathcal{A}$ in a real ANON-IBSE-CCA attack.

$\square$

C The multi-recipient case

The proofs of security given in Appendix B are easily adapted to the generalized IBSE scheme with compact multi-recipient ciphertext capability of §8.1.

We start with the the following corollary to Lemma 17, providing a generic simulation under similar restrictions as in the lemma, in the case of multi-recipient ciphertexts.

Corollary 18. *There exists an efficient algorithm $\mathcal{S}$ that, given a pair of $\mathbb{G}_1^\star$-elements $\langle g, g^\zeta \rangle$—called 'public'—, and a finite set of $\mathbb{G}_1^\star$-elements $\{g^{\xi_i}\}$—called 'crucial'—, provides an interactive simulation of all functions of a multi-recipient enabled IBSE oracle with public parameters $\langle g, g^\zeta \rangle$, when the queries are subject to the following constraints:*

1. *Never during the course of the simulation, should $\mathcal{S}$ be asked to perform either:*
   - *(a) private key extraction queries for crucial identities, i.e., any identity* id *whose public key* $H_0[\mathsf{id}]$ *belongs to the set of crucial elements $\{g^{\xi_i}\}$;*
   - *(b) decryption queries on ciphertexts signed by and encrypted for crucial identities, viz., the oracle output is unspecified on such inputs;*
   - *(c) single- or multi-recipient signature/encryption queries where the sender and at least one recipient are crucial.*
2. *At arbitrary times for as many as $\#\{g^{\xi_i}\}$ occurences throughout the simulation, $\mathcal{S}$ may be directed (by some controlling entity) to evaluate the next previously unseen random oracle query $H_0[\mathsf{id}]$ to any previously unused crucial element $g^\xi \in \{g^{\xi_i}\}$ (thereby forcibly assigning to the designated identity* id *the crucial public key g^ξ).*

During the course of a polynomial-length interaction under those conditions, $\mathcal{S}$ will be indistinguishable from a real IBSE oracle (except with some negligible probability that corresponds to the occurence of a random oracle anomaly, i.e., either a collision or the guessing of a preimage).

Proof. The proof is analogous to that of Lemma 17. We highlight the differences in the construction and the analysis.

CONSTRUCTION:

The only modifications to the construction are in the processing of multi-recipient signature/encryption queries and the decryption of multi-recipient ciphertexts, which go as follows:

- Multi-recipient signature/encryption queries: suppose $\mathcal{S}$ is to sign a message m in the name of sender id_A and encrypt it for a list of recipients $\{\mathsf{id}_{B_i} : i = 1, ..., n\}$.
  - If id_A is some guessed identity id_ξ, then $\mathcal{S}$ proceeds as follows. First, $\mathcal{S}$ picks two random numbers $r, h \in \mathbb{F}_p^\star$, lets $j = g^r\,(g^\xi)^{-h}$, and $v = (g^\zeta)^r$. Next, $\mathcal{S}$ adds the tuple $\langle j, m, h \rangle$ to the list L_1 (thereby forcing the value of the simulated random oracle to $H_1[j, m] = h$). Hence, $\mathcal{S}$ now has a signature $\langle j, v \rangle$ for the given message m and signer identity id_ξ. Then, for each recipient id_{B_i}, $i = 1, ..., n$:
    - ∗ If id_{B_i} is also a guessed identity, then $\mathcal{S}$ signals a violation of the assumptions.
    - ∗ Otherwise, $\mathcal{S}$ recovers the private key of id_{B_i}, which is given by $d_{B_i} = (g^\zeta)^{\lambda_{B_i}}$, where λ_{B_i} is found in L_0. Then, a ciphertext $\langle x_{B_i}, y_{B_i}, z_{B_i} \rangle$ is obtained by applying the function $\mathsf{EncryptToSelf}$ from Lemma 16 to the signed message $\langle \mathsf{id}_\xi, m, j, v \rangle$ using the private key d_{B_i}.

    The last step consists in aggregating the n individual ciphertexts $\langle x_{B_i}, y_{B_i}, z_{B_i} \rangle$ to form the desired multi-recipient ciphertext $\langle \langle x_{B_1}, y_{B_1} \rangle, ..., \langle x_{B_n}, y_{B_n} \rangle, z \rangle$, where z is any one of the z_{B_i} (they are all identical).
  - If instead id_A is not a guessed identity, then $\mathcal{S}$ recovers the corresponding private key $d_A = (g^\zeta)^{\lambda_A}$, where λ_A is found in L_0. Next, $\mathcal{S}$ computes $\langle j, v, m, r, \mathsf{id}_A, i_A, d_A \rangle \leftarrow \mathsf{Sign}[d_A, \mathsf{id}_A, m]$ as described in Table 1. Then, $\mathcal{S}$ computes $\langle x_{B_i}, y_{B_i}, z_{B_i} \rangle \leftarrow \mathsf{Encrypt}[\mathsf{id}_B, j, v, m, r, \mathsf{id}_A, i_A, d_A]$ for each recipient id_{B_i}, $i = 1, ..., n$. Finally, the desired multi-recipient ciphertext is assembled as $\langle \langle x_{B_1}, y_{B_1} \rangle, ..., \langle x_{B_n}, y_{B_n} \rangle, z \rangle$, where z is any one of the z_{B_i} (they are all identical).

- Decryption queries on multi-recipient ciphertext: suppose $\mathcal{S}$ is asked to decrypt a multi-recipient ciphertext $\langle\langle x_{B_1}, y_{B_1}\rangle, ..., \langle x_{B_n}, y_{B_n}\rangle, z\rangle$ for some identity id_B. First, $\mathcal{S}$ reassembles the single-recipient ciphertexts $\langle x_{B_i}, y_{B_i}, z\rangle$ for each $i = 1, ..., n$. Then:
  - If id_B is some guessed identity id_ξ, $\mathcal{S}$ proceeds to decrypt the n single-recipient ciphertexts $\langle x_{B_i}, y_{B_i}, z\rangle$ using the same procedure as in the single-recipient decryption oracle, building a list of all candidate decryptions along the way. $\mathcal{S}$ then returns any one of the resulting plaintexts, breaking ties arbitrarily (note however that the sender identity of the returned plaintext cannot be a guessed identity). If there is no eligible candidate, $\mathcal{S}$ indicates an invalid ciphertext.
  - Otherwise, $\mathcal{S}$ recovers the private key of id_B using its own key extraction oracle, and decrypts the n single-recipient ciphertexts using the regular $\mathsf{Decrypt}$ algorithm, until it finds a signed plaintext $\langle \mathsf{id}_A, m, j, v\rangle$ that is valid according to Verify. The plaintext is returned.

ANALYSIS:

We only need to study the multi-recipient functions of the simulator, since all single-recipient operations are the same as in the simulator of Lemma 17.

It is easy to observe that the multi-recipient signature/encryption oracle produces correct multi-recipient ciphertexts under the stated assumptions. In particular, since all $z_{B_i} = H_4[v] \oplus \langle \mathsf{id}_A, m\rangle$ are identical for $i = 1, ..., n$ in a given invocation of the oracle, it is legitimate to select any one of them as the z component of the multi-recipient ciphertext.

It is also easy to see that the decryption oracle gives correct answers. This is obvious in the case where the recipient identity id_B is not guessed, since then $\mathcal{S}$ has access to the private key. In the case where id_B is a guessed identity, the possible anomalies to consider are:

1. A refusal by the oracle to decrypt a valid ciphertext from a non-guessed sender identity id_A: this cannot happen, exactly by the same argument as in Lemma 17 (*i.e.*, if the ciphertext decrypts to a valid plaintext signed by id_A, then all the elements for its construction must be present on the various random oracle lists, where $\mathcal{S}$ can unambiguously find them).
2. An incorrect decryption by the oracle of the given ciphertext (regardless of its sender, or whether it is intended for id_B, or even valid). We argue that this cannot happen, because:
   - As shown in Lemma 17, each single-recipient ciphertext may lead to either zero or one candidate plaintext (from a non-guessed identity), but not more; such candidate plaintext is necessarily correct.
   - We now claim that at most one of the single-recipient ciphertexts $\langle x_{B_i}, y_{B_i}, z\rangle$, $i = 1, ..., n$, may yield a candidate plaintext (from a non-guessed identity). Suppose to a contradition that $\langle x_{B_{i_1}}, y_{B_{i_1}}, z\rangle$ and $\langle x_{B_{i_2}}, y_{B_{i_2}}, z\rangle$ respectively decrypt under id_B to the distinct valid plaintexts $\langle \mathsf{id}_{A_1}, m_1, j_1, v_1\rangle$ and $\langle \mathsf{id}_{A_2}, m_2, j_2, v_2\rangle$. First, we necessarily have $\langle \mathsf{id}_{A_1}, m_1, v_1\rangle \neq \langle \mathsf{id}_{A_2}, m_2, v_2\rangle$, since there can be only one $\langle \mathsf{id}, m, j, v\rangle$ such that $\mathsf{Verify}[\mathsf{id}, m, j, v] = \top$ when $\langle \mathsf{id}, m, v\rangle$ is fixed. Next, since $H_4[v_1] \oplus \langle \mathsf{id}_{A_1}, m_1\rangle = z = H_4[v_2] \oplus \langle \mathsf{id}_{A_2}, m_2\rangle$, it follows that $v_1 \neq v_2$, otherwise $\langle \mathsf{id}_{A_1}, m_1\rangle$ and $\langle \mathsf{id}_{A_2}, m_2\rangle$ would also have to be the same, violating our assumption that the plaintexts are distinct. Since the two plaintexts are supposedly valid, meeting the above requirements necessitates solving for $\langle \mathsf{id}_{A_1}, \mathsf{id}_{A_2}, m_1, m_2, r_1, r_2\rangle$ in the following equation:

     $$\begin{aligned} &H_4[\underbrace{H_0[\mathsf{id}_{A_1}]^{\xi\,(r_1 + H_1[H_0[\mathsf{id}_{A_1}]^{r_1}, m_1])}}_{v_1 = v_1[\mathsf{id}_{A_1}, m_1, ...]}] \oplus \langle \mathsf{id}_{A_1}, m_1\rangle \\ &= \\ &H_4[\underbrace{H_0[\mathsf{id}_{A_2}]^{\xi\,(r_2 + H_1[H_0[\mathsf{id}_{A_2}]^{r_2}, m_2])}}_{v_2 = v_2[\mathsf{id}_{A_2}, m_2, ...]}] \oplus \langle \mathsf{id}_{A_2}, m_2\rangle\ , \end{aligned}$$

     where, again, $v_1 \neq v_2$. It is clear that finding a solution to this equation requires at the very least the breaking of one of the random oracles.

We conclude that the simulation provided by $\mathcal{S}$ is accurate as long as the queries remain within the stated constraints, in the random oracle model.

□

The proofs for the five IBSE security properties (confidentiality, non-repudiation, unlinkability, authentication, anonymity) are easily adapted to the multi-recipient case, using Corollary 18.

D Dropping the irreflexivity assumption

We now restate (some of) the security results of §7 in the general case without the *irreflexivity assumption*. The irreflexivity assumption required that different identities be used for signature and encryption purposes.

We already note that the reductions are significantly more complicated and less efficient without the irreflexivity assumption. As it is particularly easy to enforce the assumption in practice, *e.g.*, by prepending all identity strings with a bit indicating their type, it is anticipated that the irreflexive mode will be the preferred mode of operation of the IBSE scheme, except in very special applications where it is crucial that signature and encryption keys be the same. These results are thus mostly of academic interest.

First, we show the following useful lemma.

Lemma 19. *Consider the following variant of the computational BDH problem:*

Given a random triple of points $\langle g, g^{\zeta}, g^{\xi}\rangle \in \mathbb{G}_1^\star$, *compute* $\mathbf{e}[g,g]^{\zeta\,\xi\,\xi} \in \mathbb{G}_2^\star$.

If there exists an algorithm $\mathcal{A}$ that solves the above problem in time τ with probability ϵ, then there exists an algorithm $\mathcal{A}'$ that solves the regular computational BDH problem in time $2\,\tau$ with probability ϵ^2.

Proof. We exhibit a construction of $\mathcal{A}'$ as follows.

Given an instance $\langle g, g^{\alpha}, g^{\beta}, g^{\gamma}\rangle$ of the computational BDH problem, $\mathcal{A}'$ combines the results of two independent calls to $\mathcal{A}$, as follows:

1. Posing $g^{\zeta_1} = g^{\alpha}$ and $g^{\xi_1} = g^{\beta}\, g^{\gamma}$ (where ζ_1 and ξ_1 are unknown), $\mathcal{A}'$ uses $\mathcal{A}$ to obtain $\mathbf{e}[g,g]^{\zeta_1\,\xi_1\,\xi_1} = \mathbf{e}[g,g]^{\alpha\,(\beta^2+\gamma^2+2\,\beta\,\gamma)}$ with probability ϵ in time τ.
2. Posing $g^{\zeta_2} = g^{\alpha}$ and $g^{\xi_2} = g^{\beta}\,(g^{\gamma})^{-1}$ (where ζ_2 and ξ_2 are unknown), $\mathcal{A}'$ uses $\mathcal{A}$ to obtain $\mathbf{e}[g,g]^{\zeta_2\,\xi_2\,\xi_2} = \mathbf{e}[g,g]^{\alpha\,(\beta^2+\gamma^2-2\,\beta\,\gamma)}$ with probability ϵ in time τ.
3. The final result, correct with probability ϵ^2, is then computed as:

$$\left(\frac{\mathbf{e}[g,g]^{\zeta_1\,\xi_1\,\xi_1}}{\mathbf{e}[g,g]^{\zeta_2\,\xi_2\,\xi_2}}\right)^{\frac{1}{4}} = \left(\frac{\mathbf{e}[g,g]^{\alpha\,(\beta^2+\gamma^2+2\,\beta\,\gamma)}}{\mathbf{e}[g,g]^{\alpha\,(\beta^2+\gamma^2-2\,\beta\,\gamma)}}\right)^{\frac{1}{4}} = \mathbf{e}[g,g]^{\alpha\,\beta\,\gamma}\ .$$

□

D.1 Confidentiality

Theorem 20. *Let $\mathcal{A}$ be a polynomial-time IND-IBSE-CCA attacker that has advantage $\geq \epsilon$, and makes $\leq \mu_i$ queries to the random oracles H_i, $i = 0,1,2,3,4$. Then, there exists a polynomial-time algorithm $\mathcal{B}$ that solves the bilinear Diffie-Hellman problem with advantage $\geq 0.15\,\epsilon^2/(\mu_0\,(\mu_1\,\mu_2+\mu_3))^2$. In the special case where encryption and signature keys are distinct, the advantage is $\geq \epsilon/(\mu_0\,\mu_2)$.*

Proof. We outline the construction of $\mathcal{B}$, that, given a BDH instance $\langle g, g^{\alpha}, g^{\beta}, g^{\gamma}\rangle$, probabilistically computes $\mathbf{e}[g,g]^{\alpha\,\beta\,\gamma}$ by interacting with $\mathcal{A}$ and emulating the required oracles. (We assume that $\mathcal{B}$ is also given the other common parameters: p, $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbf{e}$.)

We first describe a simpler algorithm, called $\mathcal{B}'$, that, given a tuple $\langle g, g^{\alpha}, g^{\zeta}, g^{\xi}\rangle$, interacts with $\mathcal{A}$ to, *either*, solve the BDH problem by computing $\mathbf{e}[g,g]^{\alpha\,\zeta\,\xi}$, *or*, solve the BDH variant defined in Lemma 19 by computing $\mathbf{e}[g,g]^{\zeta\,\xi\,\xi}$. $\mathcal{B}$ is then easily obtained by placing a wrapper around $\mathcal{B}'$, using Lemma 19.

Construction of $\mathcal{B}'$:

The construction of $\mathcal{B}'$ is based on the generic simulator described in Lemma 17, augmented as follows.

First, $\mathcal{B}'$ sets up a simulator $\mathcal{S}$ as in Lemma 17, with public parameters $\langle g, g^\zeta \rangle$ and crucial singleton $\{g^\xi\}$, where the tuple $\langle g, g^\zeta, g^\xi \rangle$ is given to $\mathcal{B}'$. In addition, $\mathcal{B}'$ picks a number η_ξ uniformly at random in $\{1, ..., \mu_0\}$. It also initializes an empty special list $L_\star$ for its own use. $\mathcal{B}'$ then runs $\mathcal{A}$ on the IBSE public parameters $\langle g, g^\zeta \rangle$, servicing all queries from $\mathcal{A}$ exactly as $\mathcal{S}$ in Lemma 17, except for the following modifications. Regarding queries to the random oracles:

- Queries to H_0:
  - If the query is the η_ξ-th distinct identifier encountered so far, then $\mathcal{B}'$ directs $\mathcal{S}$ to evaluate it as g^ξ, as per Lemma 17. The singled out queried identity is denoted id_ξ and is called the 'guessed' recipient.

As for the IBSE oracles, the modifications are:

- Signature/encryption queries:
  - If the sender identifier is id_ξ: $\mathcal{B}'$ proceeds as $\mathcal{S}$ would, per Lemma 17, which results in a (valid) signature $\langle j, v \rangle$ for the query message m and signer identity id_ξ. (Recall that this process causes the addition of a new entry $\langle j, v, h_1 \rangle \in L_1$, and involves the selection of two random numbers $r, h \in \mathbb{F}_p^\star$ to be used next (*cf.* Lemma 17).) Then:
    - If the recipient identifier is also id_ξ: instead of signaling an error as $\mathcal{S}$ would, $\mathcal{B}'$ continues as follows. $\mathcal{B}'$ successively picks two random elements $u \in \mathbb{G}_2^\star$ and $w \in \mathbb{G}_2^\star$, obtains $h_2 = H_2[w]$, $h_3 = H_3[u]$, $h_4 = H_4[v]$, and computes $x = j^{h_3}$, $y = h_2 \oplus v$, $z = h_4 \oplus \langle \mathsf{id}_\xi, m \rangle$. Finally, $\mathcal{B}'$ adds the tuple $\langle \mathsf{id}_\xi, m, j, v, u, w, r, h \rangle$ to the special list $L_\star$, and returns the (fake) ciphertext $\langle x, y, z \rangle$.
    - Otherwise, $\mathcal{B}'$ proceeds as $\mathcal{S}$ in the case $\mathsf{id}_A = \mathsf{id}_\xi \neq \mathsf{id}_B$, *i.e.*, computing the private key of the recipient and using $\mathsf{EncryptToSelf}$ to produce a (correct) ciphertext from the signed message obtained thus far. In addition, $\mathcal{B}'$ adds the tuple $\langle \mathsf{id}_\xi, m, j, v, u, w, r, h \rangle$ to the special list $L_\star$, where u and w are computed during the execution of $\mathsf{EncryptToSelf}$ (*cf.* Lemma 16).

  Note that a tuple $\langle \mathsf{id}_\xi, m, j, v, u, w, r, h \rangle$ is added to the special list $L_\star$ in all cases where the sender identity is id_ξ.

- Decryption queries:
  - If the recipient identity is id_ξ, $\mathcal{B}'$ proceeds exactly as $\mathcal{S}$ would, as in Lemma 17, except for the following modifications:
    - If $\mathcal{S}$ produces a decrypted plaintext $\langle \mathsf{id}_A, m, j, v \rangle$ with $\mathsf{id}_A \neq \mathsf{id}_\xi$, then $\mathcal{B}'$ simply returns it.
    - In case $\mathcal{S}$ returns an error, *i.e.*, fails to find a valid decryption candidate $\langle \mathsf{id}_A, m, j, v \rangle$ with $\mathsf{id}_A \neq \mathsf{id}_\xi$, then $\mathcal{B}'$ continues as follows. First, $\mathcal{B}'$ searches the lists maintained by $\mathcal{S}$ for all instances of $\langle \mathsf{id}_\xi, m, j, v \rangle$ such that $\langle j, m, h_1 \rangle \in L_1$, $\langle w, h_2 \rangle \in L_2$, $\langle u, h_3 \rangle \in L_3$, $\langle v, h_4 \rangle \in L_4$, for some h_1, h_2, h_3, h_4, u, w, under the constraints that $h_2 \oplus y = v$, $x^{h_3^{-1}} = j$, $h_4 \oplus z = \langle \mathsf{id}_\xi, m \rangle$, $\mathsf{Verify}[\mathsf{id}_\xi, m, j, v] = \top$, and $\langle \mathsf{id}_\xi, m, j, v, u, w \rangle \in L_\star$. Then, $\mathcal{B}'$ returns any tuple $\langle \mathsf{id}_\xi, m, j, v \rangle$ satisfying all of the above conditions (breaking ties arbitrarily). If no such tuple is found, $\mathcal{B}'$ indicates that the ciphertext is invalid.

- Key extraction queries:
  - If the key extraction query identity is id_ξ, then $\mathcal{B}'$ terminates its interaction with $\mathcal{A}$, having failed to guess the targeted recipient among those in L_0.

At some point, $\mathcal{A}$ will produce two identities id_A and id_B and two equal-length messages m_0 and m_1 on which it wishes to be challenged. $\mathcal{B}'$ responds with a challenge ciphertext $\langle g^\alpha, y, z \rangle$, where y and z are random strings of appropriate size.

All further queries by $\mathcal{A}$ (subject to the appropriate additional restrictions) are processed as previously described.

Finally, $\mathcal{A}$ returns its final guess. $\mathcal{B}'$ ignores it, and produces its own answer in the following randomized manner.

Let $\phi_{\text{BDH}} \in \{r_{\text{BDH}}, v_{\text{BDH}}\}$ be a binary flag, where the value r_{BDH} signifies an attempt at solving the regular BDH problem, and the value v_{BDH} indicates an attempt at solving the BDH variant of Lemma 19. Then, for some choice probabilities $z_1, z_2, z_3 \geq 0$ with $z_1 + z_2 + z_3 = 1$, to be determined later:

Choice 1 With probability z_1, $\mathcal{B}'$ picks an entry $\langle \bar{w}, h_2 \rangle$ uniformly at random in L_2, and returns the tuple $\langle r_{\text{BDH}}, \bar{w} \rangle$.

Choice 2 With probability z_2, $\mathcal{B}'$ picks an entry $\langle \bar{u}, h_3 \rangle$ uniformly at random in L_3, and returns the tuple $\langle v_{\text{BDH}}, \bar{w} \rangle$.

Choice 3 With probability z_3, $\mathcal{B}'$ picks an entry $\langle \bar{w}, h_2 \rangle$ in L_2, and an entry $\langle \mathsf{id}, m, j, v, u, w, r, h \rangle$ in $L_\star$, uniformly at random such that $\mathsf{id} = \mathsf{id}_\xi$ and $\exists h_3 : \langle u, h_3 \rangle \in L_3$; it then returns the tuple $\langle v_{\text{BDH}}, \bar{w}^{-(h\, h_3)^{-1}}\, \mathbf{e}[g^\xi, g^\zeta]^{h^{-1}\, r} \rangle$.

Observe that the first case corresponds to a random choice by $\mathcal{B}'$ to try and solve the regular BDH problem, as indicated by the flag $\phi_{\text{BDH}} = r_{\text{BDH}}$; the last two cases correspond to attemps by $\mathcal{B}'$ to solve the BDH variant of Lemma 19, as indicated by $\phi_{\text{BDH}} = v_{\text{BDH}}$.

CONSTRUCTION OF $\mathcal{B}$:

We now show how to construct $\mathcal{B}$, using as a subroutine the machine $\mathcal{B}'$ that is itself interacting with $\mathcal{A}$.

Given a BDH instance $\langle g, g^\alpha, g^\beta, g^\gamma \rangle$, $\mathcal{B}$ makes three independent calls to $\mathcal{B}'$ with different input parameters, as follows:

- $\mathcal{B}'$ is invoked on the input tuple $\langle g, g^\alpha, g^\zeta, g^\xi \rangle$ with $g^\zeta = g^\gamma$ and $g^\xi = g^\beta$ (where ζ and ξ are of course unknown). Denote by $\langle \phi_{\text{BDH}1}, e_1 \rangle$ the outcome of this first execution of $\mathcal{B}'$.
- $\mathcal{B}'$ is invoked on the input tuple $\langle g, g^\alpha, g^\zeta, g^\xi \rangle$ with $g^\zeta = (g^\alpha)^{r_2}$ and $g^\xi = g^\beta\, g^\gamma$, where r_2 is taken at random in $\mathbb{F}_p^\star$. Denote by $\langle \phi_{\text{BDH}2}, e_2 \rangle$ the outcome of this second execution of $\mathcal{B}'$.
- $\mathcal{B}'$ is invoked on the input tuple $\langle g, g^\alpha, g^\zeta, g^\xi \rangle$ with $g^\zeta = (g^\alpha)^{r_3}$ and $g^\xi = g^\beta\, (g^\gamma)^{-1}$, where r_3 is taken at random in $\mathbb{F}_p^\star$. Denote by $\langle \phi_{\text{BDH}3}, e_3 \rangle$ the outcome of this third execution of $\mathcal{B}'$.

$\mathcal{B}$ then synthesizes its guess for the value of $\mathbf{e}[g, g]^{\alpha\,\beta\,\gamma}$, as follows:

- If $\phi_{\text{BDH}1} = r_{\text{BDH}}$, then $\mathcal{B}$ returns e_1.
- Else, if $\phi_{\text{BDH}2} = v_{\text{BDH}}$ and $\phi_{\text{BDH}3} = v_{\text{BDH}}$, then $\mathcal{B}$ returns $(e_2^{r_2^{-1}} / e_3^{r_3^{-1}})^{1/4}$.
- Otherwise, $\mathcal{B}$ declares failure.

ANALYSIS:

We first focus our analysis on $\mathcal{B}'$.

Per Lemma 17, it is easy to see that the simulation provided by $\mathcal{B}'$ is true to a genuine attack scenario, except in the following respects:

1. The challenge ciphertext eventually presented to $\mathcal{A}$: the method employed by $\mathcal{B}'$ to construct the ciphertext almost certainly results in an invalid challenge.
2. Signature/encryption queries when sender and recipient identities are both set to id_ξ: the oracle is unable to construct a correct ciphertext in this case, and returns an incorrect one instead.
3. Decryption queries for the recipient id_ξ for which the only plaintext candidates all bear valid signatures also from id_ξ: in this case the oracle always rejects the ciphertext, even if it is valid, unless it can be positively traced (via the simulated random oracles and the special list $L_\star$) to the result of a preceding signature/encryption query.

Notice that there is no risk that the decryption oracle erroneously decrypt an incorrect ciphertext for recipient id_ξ, by virtue of the verification performed against the special list[5].

Since the challenge ciphertext provided by $\mathcal{B}'$ is randomly distributed in the space of ciphertexts of correct size, there is no way that $\mathcal{A}$ could gain any advantage in this simulation. Thus, any adversary that has advantage ϵ in the real IND-IBSE-CCA game must recognize with probability at least ϵ that the simulation provided by $\mathcal{B}'$ is incorrect.

Now, there are only three types of anomalies in the simulation by $\mathcal{B}'$, the detectability of each of which by $\mathcal{A}$ leads to a polynomial-time solution of the BDH problem or its variant by $\mathcal{B}'$:

1. Recognition that the challenge ciphertext of the form $\langle x, y, z\rangle$ with $x = g^\alpha$ is invalid: this requires a random oracle query $H_2[\bar{w}]$ with $\bar{w} = \mathbf{e}[x, d_\xi] = \mathbf{e}[g, g]^{\alpha\,\zeta\,\xi}$. Any such query by $\mathcal{A}$ leaves an entry $\langle \bar{w}, h_2\rangle$ on L_2, from which $\mathcal{B}'$ can then extract $\mathbf{e}[g, g]^{\alpha\,\zeta\,\xi} = \bar{w}$ with (conditional) probability $1/\mu_2$.
2. Decryption queries for id_ξ where all valid plaintext candidates are signed by id_ξ: recognizing that such a ciphertext $\langle x, y, z\rangle$ is in fact correct even though it is rejected by the decryption oracle, requires a random oracle query $H_3[\bar{u}]$ with $\bar{u} = \mathbf{e}[i_\xi, d_\xi] = \mathbf{e}[g, g]^{\zeta\,\xi\,\xi}$. Any such query by $\mathcal{A}$ leaves an entry $\langle \bar{u}, h_3\rangle$ on L_3, from which $\mathcal{B}'$ can extract $\mathbf{e}[g, g]^{\zeta\,\xi\,\xi} = \bar{w}$ with (conditional) probability $1/\mu_3$.
3. Signature/encryption queries when sender and recipient identities are both id_ξ: in this case the ciphertext returned by the oracle is always of the form $\langle x, y, z\rangle$ where $x = (g^r\,(g^\xi)^{-h})^{h_3}$ for some $\langle \mathsf{id}_\xi, m, j, v, u, w, r, h\rangle \in L_\star$ with $\langle u, h_3\rangle \in L_3$. Recognizing that such a ciphertext is invalid requires a random oracle query $H_2[\bar{w}]$ with $\bar{w} = \mathbf{e}[x, d_\xi] = \mathbf{e}[g^\xi, g^\zeta]^{r\,h_3}\,\mathbf{e}[g, g]^{-h\,h_3\,\zeta\,\xi\,\xi}$. Any such query made by $\mathcal{A}$ leaves an entry $\langle \bar{w}, h_2\rangle$ on L_2, from which $\mathcal{B}'$ can extract

   $$\mathbf{e}[g, g]^{\zeta\,\xi\,\xi} = \bar{w}^{-(h\,h_3)^{-1}}\,\mathbf{e}[g^\xi, g^\zeta]^{h^{-1}\,r}\ ,$$

   with (conditional) probability $1/(\mu_2\,|L_\star|) \le 1/(\mu_1\,\mu_2)$, since the list $L_\star$ is never larger than L_1.

We now determine the total probability of $\mathcal{B}$ solving the original BDH problem. Doing so, we also determine the choice probabilities z_1, z_2, z_3, left open in the construction of $\mathcal{B}'$.

Let ϵ_1, ϵ_2, and ϵ_3 be the respective probabilities that $\mathcal{A}$ encounters at least one simulation anomaly of the first, second, and third type, during the course of an interaction with $\mathcal{B}'$. Taking into account the probability $1/\mu_0$ of $\mathcal{B}'$ making the correct choice for the guessed identity id_ξ, the total probability that $\mathcal{B}$ solves the BDH problem is given by:

$$\begin{aligned}\mathbf{Adv}[\mathcal{B}] &\equiv \mathbf{P}[\mathbf{e}[g, g]^{\alpha\,\beta\,\gamma} \leftarrow \mathcal{B}[g, g^\alpha, g^\beta, g^\gamma]] \\ &= \left(\frac{z_1}{\mu_0\,\mu_2}\,\epsilon_1\right) + (1 - z_1)\,\left(\frac{z_2}{\mu_0\,\mu_3}\,\epsilon_2 + \frac{z_3}{\mu_0\,\mu_1\,\mu_2}\,\epsilon_3\right)^2\ .\end{aligned}$$

Let $\rho \approx 0.68$ be the real root of $\rho^3 + \rho - 1 = 0$. Let $z_1 = 1 - \rho$, $z_2 = \rho\,\frac{\mu_3}{\mu_1\,\mu_2 + \mu_3}$, $z_3 = \rho\,\frac{\mu_1\,\mu_2}{\mu_1\,\mu_2+\mu_3}$, noting that $z_1 + z_2 + z_3 = 1$. The advantage of $\mathcal{B}$ becomes:

$$\mathbf{Adv}[\mathcal{B}] = (1 - \rho)\,\left(\left(\frac{\epsilon_1}{\mu_0\,\mu_2}\right) + \left(\frac{\epsilon_2 + \epsilon_3}{\mu_0\,(\mu_1\,\mu_2 + \mu_3)}\right)^2\right)\ .$$

Clearly, the advantage of $\mathcal{A}$ must satisfy $\mathbf{Adv}[\mathcal{A}] \le \epsilon_1 + \epsilon_2 + \epsilon_3$. Note however that we cannot make any assumption regarding the relative sizes of ϵ_1, ϵ_2, and ϵ_3; hence, $\mathcal{B}$ should provide a reduction that is uniformly efficient for all three possible types of anomalies. Hence, considering the worst case over $0 \le \epsilon_1, \epsilon_2, \epsilon_3 \le 1$

[5]The role of the special list $L_\star$ is to ensure, in the decryption oracle, that the values $\langle u, w\rangle$ associated with a candidate plaintext $\langle \mathsf{id}_\xi, m, j, v\rangle$ are those associated with that plaintext in a previous call to the signature/encryption. This ensures that, among all candidate plaintexts with a valid signature from id_ξ, exactly those that have been constructed in a previous call to the signature/encryption oracle, and no other, are accepted by the decryption oracle (barring a negligible probability of random oracle collision).

such that $\mathbf{Adv}[\mathcal{A}] \leq \epsilon_1 + \epsilon_2 + \epsilon_3$, noting that $\epsilon_1 + (\epsilon_2 + \epsilon_3)^2 \geq (\epsilon_1 + \epsilon_2 + \epsilon_3)^2/2$, we get:

$$\begin{aligned} \mathbf{Adv}[\mathcal{B}] &\geq (1-\rho) \left(\frac{\epsilon_1 + (\epsilon_2 + \epsilon_3)^2}{(\mu_0\,(\mu_1\,\mu_2 + \mu_3))^2} \right) \\ &\geq \frac{1-\rho}{2} \left(\frac{\mathbf{Adv}[\mathcal{A}]}{\mu_0\,(\mu_1\,\mu_2 + \mu_3)} \right)^2 \qquad \approx 0.15 \left(\frac{\mathbf{Adv}[\mathcal{A}]}{\mu_0\,(\mu_1\,\mu_2 + \mu_3)} \right)^2 . \end{aligned}$$

This proves the general case of the theorem.

Special case with distinct sender and recipient identities (Theorem 11):

In the special case where the sender and recipient identities and keys are distinct, the two types of simulation anomalies involving $\mathsf{id}_A = \mathsf{id}_\xi = \mathsf{id}_B$ disappear, and the $\mathcal{B}'$ simulation becomes perfect in all regards except for the ciphertext challenge eventually given to $\mathcal{A}$. In this situation, $\mathbf{Adv}[\mathcal{A}] \geq \epsilon_1$.

We modify $\mathcal{B}'$ to take Choice 1 with probability $z_1 = 1$, and reprogram $\mathcal{B}$ to simply invoke $\mathcal{B}'$ without any additional processing. In this case, the expression of $\mathbf{Adv}[\mathcal{B}]$ in terms of $\mathbf{Adv}[\mathcal{A}]$ reduces to:

$$\mathbf{Adv}[\mathcal{B}] \geq \frac{1}{\mu_0\,\mu_2}\,\mathbf{Adv}[\mathcal{A}] \ ,$$

which is another proof of the special case considered in Theorem 11. This concludes the proof of Theorem 20. $\square$

Errata

The following corrections should be made to the version appearing in CRYPTO '03:

In §6, Table 1: in the description of Verify, the expression "$H_1[\hat{j}, m]$" should be read as "$H_1[\hat{j}, \hat{m}]$".

In §6, Table 2: the plaintext sizes for "IB E-then-S" and "IB AE-then-S" should be ignored.

In §6, Table 2: a few numerical entries should be replaced by the values in this version.

In §8.2: the text fragment "substituting g^σ for $\hat{i}_A$" should be read as "substituting g for $\hat{i}_A$".

The following additional changes were made to this paper since it first appeared online:

[2003.08.25] Fixed page numbers in citation header, added eprint reference.

[2004.02.25] Fixed some formatting issues; clarified the legend of Table 2.