Scientific Research Publishing

Journal of Information Security, 2022, 13, 323-336
https://www.scirp.org/journal/jis
ISSN Online: 2153-1242
ISSN Print: 2153-1234

Security Analysis of a Privacy-Preserving Identity-Based Encryption Architecture

Carlisle Adams

School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, Canada
Email: cadams@uottawa.ca

How to cite this paper: Adams, C. (2022) Security Analysis of a Privacy-Preserving Identity-Based Encryption Architecture. *Journal of Information Security*, **13**, 323-336.
https://doi.org/10.4236/jis.2022.134018

Received: August 5, 2022
Accepted: October 9, 2022
Published: October 12, 2022

Abstract

Identity-Based Encryption (IBE) has seen limited adoption, largely due to the absolute trust that must be placed in the private key generator (PKG)—an authority that computes the private keys for all the users in the environment. Several constructions have been proposed to reduce the trust required in the PKG (and thus preserve the privacy of users), but these have generally relied on unrealistic assumptions regarding non-collusion between various entities in the system. Unfortunately, these constructions have not significantly improved IBE adoption rates in real-world environments. In this paper, we present a construction that reduces trust in the PKG without unrealistic non-collusion assumptions. We achieve this by incorporating a novel combination of digital credential technology and bilinear maps, and making use of multiple randomly-chosen entities to complete certain tasks. The main result and primary contribution of this paper are a thorough security analysis of this proposed construction, examining the various entity types, attacker models, and collusion opportunities in this environment. We show that this construction can prevent, or at least mitigate, all considered attacks. We conclude that our construction appears to be effective in preserving user privacy and we hope that this construction and its security analysis will encourage greater use of IBE in real-world environments.

Keywords

Security Analysis, Identity-Based Encryption (IBE), Reducing Trust, Preserving Privacy, Honest-but-Curious Attacker, Malicious Attacker

1. Introduction

In 1984, Shamir proposed the concept of identity-based cryptography (including both Identity-Based Encryption (IBE) and Identity-Based Signatures (IBS) [1]),

in which a user's identity (for example, a user's e-mail address) can effectively be used as a public key for cryptographic purposes. An efficient scheme to realize IBE was proposed in 2001 by Boneh and Franklin [2] (see also [3]). In this construction (and in subsequent constructions by other researchers), a private key generator (PKG) computes all user private keys upon request.

IBE is a very interesting and promising technology that has unfortunately seen limited adoption and deployment, in large part because of the requirement for complete trust in the PKG (since the PKG knows the private keys of all users). Schemes for reducing trust in the PKG have been proposed by a number of researchers (see, for example, [2] [3] [4] [5] [6]), but have generally relied on assumptions that do not match real-world environments (for example, there are two specified entities, and it must be guaranteed that these two entities will never collude (in perpetuity), even if they exist within a single company).

In this paper, we present a scheme to reduce trust in the PKG that does not require unrealistic assumptions about non-collusion. In our scheme, there are multiple entities, and collusion will only lead to a successful attack if a randomly-chosen collection of these entities are all malicious. Thus, the risk of a successful attack can be reduced to an arbitrarily small level by increasing the number of entities that must participate in a given task. Naturally, this increases the amount of computation for a task, but not prohibitively: administrators can trade-off computation against the risk of collusion in any deployment so that an appropriate balance can be achieved for the environment.

We provide an extensive security analysis of our scheme, showing the protection that it delivers against a wide variety of attacker models. The ultimate goal of our construction and security analysis is to re-ignite interest in IBE and to hopefully increase the deployment of IBE in real-world settings.

2. Previous Work

A number of techniques have been proposed over the years to reduce the trust in the PKG (sometimes referred to as techniques to eliminate, or at least mitigate, the escrow problem—since the PKG may retain a copy of each user's private key—or as techniques to improve user privacy in IBE deployments—since a rogue PKG may decrypt and read Alice's ciphertext without her consent). These techniques have typically been either threshold schemes or separation schemes. In a threshold scheme, there are n PKGs and a subset of at least τ of them is required in order to compute a given user's private key (see, for example, Boneh and Franklin [2] [3] and Bendlin *et al.* [5]). In a separation scheme, on the other hand, the PKG is split into an intermediate certification authority (ICA) that verifies the user's identity and generates a blinded token for the user, and a PKG that receives a blinded token and generates the appropriate blinded private key which the user can then unblind (see, for example, Chow [4] and Emura *et al.* [6]).

Unfortunately, threshold schemes and separation schemes often rely on

non-collusion assumptions that are difficult or impossible to guarantee with certainty in real-world environments. In particular, threshold schemes require that there are never more than n-τ malicious parties in the environment (and, more importantly, that there will never be τ malicious parties that collude to learn Alice's private key). Separation schemes require that the PKG and the ICA will never collude during the lifetime of the deployment, even if they are personnel working in the same company.

Because the underlying assumptions of most threshold and separation schemes are difficult to guarantee in practice, IBE continues to see limited deployment and use in real environments. Our proposal is also a separation scheme, but is designed in such a way that the risk of a successful collusion attack can be made arbitrarily small in a real-world setting.

3. Our Proposal

Due to space constraints, our proposal is described in significantly more detail in a companion paper[1]. However, a brief description is given in this section to set a foundation for understanding the security analysis presented in Section 4. We begin in Section 3.1 by listing the background technologies used in our construction, and then outline the construction itself in Section 3.2.

3.1. Background Technologies

The technologies used in our construction include digital credentials, elliptic curves, bilinear maps, and the FullIdent IBE algorithm in [2] [3].

Digital Credentials. In many separation schemes, the blinded token that the ICA gives to the user after the identity has been verified (and that the user subsequently gives to the PKG to obtain a blinded private key) is typically some form of a public key certificate. In our construction, this token is in the form of a digital credential as designed by Brands [7] [8]. Using the digital credential form allows the possibility of an arbitrary number of pseudonymization steps, meaning that for the PKG to learn Alice's identity (and therefore learn her private key), every pseudonymizing ICA in the chain would have to be malicious.

Elliptic Curves. For carefully-chosen curves and parameters, elliptic curve groups are believed to provide high levels of security with much smaller keys than are required for comparable cryptographic operations over multiplicative groups [9] [10]. Elliptic curve groups are therefore used as the basis for some cryptographic algorithms, including the IBE algorithm proposed by Boneh and Franklin [2] [3].

Bilinear Maps. A bilinear map (or bilinear pairing) is a function $\hat{e}(\cdot, \cdot)$ that maps from one group of elements of prime order q (G_1) to another group of elements also of order q (G_2) where the mapping is bilinear, non-degenerate, and efficiently computable [11] [12]. Specifically, for $\hat{e}$: $G_1 \times G_1 \rightarrow G_2$, the mapping is bilinear if $\forall\ P_1, P_2 \in G_1$ and $\forall\ a, b \in Z_q^*$, $\hat{e}(aP_1, bP_2) = \hat{e}(P_1, P_2)^{ab}$, and it is

[1]Adams, C. (2022) Improving user privacy in identity-based encryption deployments.

non-degenerate if ∀ non-trivial points $P_1 \in G_1$, $\hat{e}(P_1, P_1) \neq 1$ (the multiplicative identity element in G_2). In many instantiations (including the one we use in our construction), G_1 is an elliptic curve group of order q (generated by a known generator point G) and G_2 is the group of integers modulo q.

IBE. Our construction uses the FullIdent IBE algorithm from Boneh and Franklin [2] [3]. This algorithm encrypts an n-bit plaintext to a 3-component ciphertext $c = (u, v, w)$ under a public key which is a function H_1 of a user's identity (*i.e.* the user's e-mail address). The security of this algorithm is proved in [3] (see also Section 4.4 below).

3.2. Proposed Construction

As with some previous proposals (such as [4] [6]), our construction is a separation scheme, separating the role of the ICA from the role of the PKG. However, unlike those proposals, we specify multiple ICA entities (ICA_1, …, ICA_v) where each ICA can provide both a *credentialing service* and a *pseudonymizing service*. For the credentialing service, an ICA inputs a user identity (an e-mail address) and outputs a digital credential for the attribute $a_1 = \hat{e}(P, P)$, where $P = H_1$(identity). For the pseudonymizing service, an ICA inputs a digital credential, a point Q, and a random value s; it verifies that the attribute a_1 in the submitted credential is $\hat{e}(Q, Q)$ and, if so, outputs a new digital credential (a pseudonym) for the attribute $a_1 = \hat{e}(sQ, sQ)$. The user now has a randomized/blinded version of the original credential. This pseudonymizing step can be done any number of times with different randomly-chosen ICAs (and different random values s_i). The final pseudonym can be submitted to the PKG to obtain a blinded version of the private key, which the user can easily unblind because the user knows all the random s_i that were used in the chain of pseudonymizations.

The credential and the pseudonym have an identical form: $h = (g_1^{a1} \cdot g_2^{a2} \cdot h_0)^{\alpha} \bmod p$, where g_1, g_2, and h_0 are generators of the q-order subgroup of Z_p^* (these values are system parameters known to everyone), a_2 is the unique identifier of the original ICA that has signed this credential/pseudonym, p is a publicly-known prime modulus, and α is a private value known only to the user that owns this credential. (This formula for h is the standard format specified by Brands for digital credentials; it provides unconditional hiding of any values not explicitly shown to a verifier (in this case, the private key α). Digital credentials are used in our construction as described above[2]; see [8] for further details on digital credentials.)

Note that (from attribute a_2) any given pseudonymizing ICA (say ICA_k) knows only the original ICA that signed the credential it received (say ICA_j); it does not know any ICAs that produced a pseudonym earlier in the chain. Therefore, for the PKG to learn Alice's identity, it will need to collude with *every* ICA in the chain of pseudonyms (*i.e.* all the way back to the credentialing ICA) and so this collusion attack will only be successful if *every* ICA in the chain is malicious.

[2]Ibid.

Given that the user has chosen these ICAs randomly, the risk that they are all malicious can be made as small as the user desires by picking an appropriate chain length. (In particular, if the probability that an ICA is malicious is ρ, then the probability that the PKG will learn Alice's identity through collusion with the pseudonymizing ICAs is ρ^z if Alice chooses her chain to be of length z.)

4. Security Analysis

This section provides a security analysis of the construction given in Section 3.2. We begin by describing the scenario (*i.e.* the context) for our analysis, the attackers and attacker model that we assume, and the security goal that we wish to achieve. We then look at the security of the encryption algorithm itself and security against various collections of entities in this environment.

4.1. Scenario

The scenario that forms the context for our security analysis is simple and fundamental: some plaintext data has been encrypted for user Alice in an IBE environment. To avoid trivial attacks, we stipulate that the resulting ciphertext may be visible to other parties (for example, it may be stored in a publicly-accessible database, web server, or cloud service), but it is not explicitly labeled. In particular, another party seeing the ciphertext does not know that this ciphertext is intended for Alice. (An alternative way of describing this constraint is that the ciphertext **is** labeled with a name or other unique identifier for Alice, but there is no obvious mapping from this label to Alice's e-mail address (which corresponds to the public key under which this data was encrypted for her), so that an observer of this ciphertext does not know which public key was used to encrypt it.)

4.2. Attackers and Attacker Model

Aside from Alice, there are 4 types of entities in the environment: other users (e.g. Bob); credentialing ICAs; pseudonymizing ICAs; and the PKG.

An entity acting as an attacker may be modeled as *honest-but-curious* (*HbC*) or as *malicious*. An entity that is *HbC* can be expected to faithfully comply with all required algorithms and communication protocols, but will privately do whatever it can (beyond this) to learn information that it is not supposed to know. In particular, it may

- Store transaction data or protocol metadata and perform offline computation and analysis to draw inferences or determine sensitive information.

An entity that is *malicious* is similar to an *HbC* entity, except that it cannot be expected to faithfully comply with algorithms and protocols. In particular, it may

- Respond to (arbitrary) queries from other parties for the transaction data or protocol metadata it has stored if this will help these other parties to draw inferences or determine sensitive information (and share the results).
- Create cryptographically invalid data structures and constructs.

- Omit/skip/overlook the validation of cryptographic protections on data structures and constructs created by others.

We assume that any of the 4 types of entities listed above may be *HbC* or *malicious*, and that collusions among any collection of malicious entities is possible. Such a model is reasonable and realistic because it includes entities that work alone (and try to hide their nefarious activities by following all specified algorithms and protocols in their interactions with others), as well as entities that will collude with other parties (using unauthorized algorithms and protocols). Thus, the security analysis based on this model will encompass both hidden and overt attackers, which matches what may be found in real environments.

In addition, we assume that all entities in the environment are polynomially-bounded; that is, we assume that they have time, memory, and computational resources that are bounded by a polynomial in the security parameter of the IBE encryption algorithm. (This assumption is common in security analyses of cryptographic algorithms and protocols: if the attackers are instead assumed to have super-polynomial capabilities, many algorithms and protocols will be unable to provide any hope of security.)

4.3. Security Goal

The security goal for this environment is the following: *no entity other than Alice is able to learn her private key and/or decrypt the ciphertext that was intended for her*. More specifically, the focus of our analysis in the following subsections is to understand the precise conditions under which this security goal may be achieved.

4.4. Security of the Encryption Algorithm

The most obvious (though seldom the simplest) way for an attacker to learn Alice's private key, or to decrypt the ciphertext that was intended for her, is to break the encryption algorithm itself. Clearly, if the algorithm can be broken, then the attacker can learn something about the plaintext from the ciphertext, or can recover the private key from the public key and/or the ciphertext.

The underlying encryption algorithm used in our proposal is exactly the "FullIdent" scheme given in Section 4.2 of Boneh and Franklin [3]. This algorithm is proved in that paper to be an *adaptive chosen ciphertext secure* (*IND-ID-CCA*) *IBE in the random oracle model*, assuming that the Bilinear Diffie-Hellman (BDH) problem is hard in groups generated by a BDH parameter generator $\mathcal{G}$ (that is, a generator $\mathcal{G}$ that, on input a security parameter k, generates a prime q, two groups G_1, G_2 of order q, and an admissible bilinear map $\hat{e}$: $G_1 \times G_1 \rightarrow G_2$).

Thus, no polynomially-bounded (probabilistic polynomial time, PPT) adversary $\mathcal{A}$ has a non-negligible advantage against a challenger in a formally-defined IND-ID-CCA game (see [3] for the proof details), meaning that an active PPT adversary can learn nothing about the plaintext from the ciphertext (this includes learning nothing about the full private key, because learning the private

key necessarily implies learning the plaintext). Therefore, the ciphertext is secure from every entity in the environment who may try to learn the private key or plaintext by breaking the underlying encryption algorithm.

4.5. Security against HbC Entities

By definition, *honest-but-curious* entities do not collude to learn information they are not supposed to know. Because such entities faithfully comply with all required algorithms and communication protocols, all interactions with any other entity use the messages and data structures specified for the environment and so there is no opportunity to establish collusion for nefarious purposes. Thus, an HbC entity works alone with its own stored data, computational resources, and ingenuity.

HbC User. An honest-but-curious user Bob wishes to learn Alice's private key so that he can decrypt ciphertext that was encrypted for her. Because Bob is restricted to legitimate messages and data structures, he must obtain a valid credential that he can present to the PKG (to acquire Alice's private key), which means that he must successfully impersonate Alice when he interacts with the credentialing ICA. Therefore, Bob must convince the credentialing ICA that he is the owner of Alice's ID (*i.e.* her e-mail address). Bob needs to know Alice's precise e-mail address and he needs a way of proving ownership of that address.

If proof of ownership is done, for example, by Bob demonstrating the ability to send and receive e-mails using that address, then protection against an HbC Bob rests only on the difficulty of Bob learning Alice's Gmail password (say).

Note that authentication to the ICA credentialing service (that is, proving ownership of a claimed identity) is outside the scope of IBE algorithms and protocols. However, this is clearly an essential component in the overall security of the system. Proof of ownership of an identity must be sufficiently trustworthy that it precludes impersonation attacks by other entities.

HbC PKG. An honest-but-curious private key generator wishes to compute Alice's private key so that it can decrypt ciphertext that was encrypted for her. The PKG receives a pseudonymized credential and a randomized point $R = (s_z \ldots s_1)P$ (where $P = H_1(\text{identity})$), and it returns the key $K = tR = t(s_z \ldots s_1)P$. However, it would like to learn $K_A = tP$. Security is provided by the elliptic curve discrete logarithm problem if R and P are known, but in this case the PKG also does not know P (it only knows the generator point G). Therefore, a polynomially-bounded PKG will not be able to learn K_A if elliptic curve parameters of sufficient size are chosen for the implementation of this scheme.

Note that if the HbC PKG is able to learn P by any means, then it can easily compute $K_A = tP$. In a closed-world setting (*i.e.* one in which the PKG knows all possible users in the environment), the PKG can compute each user's private key to see which one decrypts a given target ciphertext. Such an attack cannot be prevented. (One potential avenue is for the encrypter to add entropy in some way to the encryption process, but then these random bits would need to be se-

curely conveyed to Alice so that she could decrypt, which clearly deviates from a pure IBE environment.)

On the other hand, in an open-world setting (where users are members of the general public), the PKG cannot know all possible users and their corresponding e-mail addresses. This will protect against an HbC PKG attack, but only if the size of the identity space is sufficiently large that the PKG cannot exhaustively try all identities.

Having a sufficiently large identity space in an open-world setting is outside the scope of IBE algorithms and protocols. However, this is also an essential component in the overall security of the system.

HbC psedonymizing ICA. An honest-but-curious pseudonymizing intermediate CA wishes to learn Alice's private key so that it can decrypt ciphertext that was encrypted for her. The ICA receives a pseudonym, a randomized elliptic curve point $(s_{i-1}\ldots s_1)$P and a random value s_i, and it returns a pseudonym for the attribute $a_1 = \hat{e}((s_i\ldots s_1)P, (s_i\ldots s_1)P)$. Alternatively, it learns P and constructs its own $(s_i\ldots s_1)$P and corresponding pseudonym. It should be clear that this is identical to the HbC PKG attack above. The elliptic curve discrete logarithm problem provides security against one form of the attack, and an open-world setting with a sufficiently large identity space provides security against the other form of the attack.

Again, choosing suitable elliptic curve parameters and having an open-world setting with a large identity space are essential components in achieving system security.

HbC credentialing ICA. An honest-but-curious credentialing intermediate CA wishes to learn Alice's private key so that it can decrypt ciphertext that was encrypted for her. If the ICA credentialing service has been invoked by Alice, then the ICA will be given Alice's identity (*i.e.* her e-mail address) and will be able to compute $P = H_1$ (identity). The ICA can then create its own credential for P (*i.e.* using its own value for α and not interacting with Alice at all); the ICA can then pseudonymize this credential—note that pseudonymizations must take place over an anonymous channel such as Tor so that the requesting user's identity remains hidden; therefore the pseudonymizing ICA will not know that it is dealing with the credentialing ICA instead of an actual user—and then interact with the PKG to obtain Alice's private key.

The attack just described is precisely the reason why the ICA is stipulated not to have any interaction with the PKG in the formal security analysis of Emura *et al.* [6]. Interestingly, this same vulnerability has always existed in traditional public key infrastructure (PKI; see [13] [14]) environments: a rogue CA can generate its own key pair and create a public key certificate with Alice's name and this generated public key. Other users may then trust this certificate and encrypt data intended for Alice using the contained public key (Alice will of course not be able to decrypt the resulting ciphertext, but the rogue CA will be able to decrypt it).

The credentialing ICAs can be implemented as smart contracts [15] [16] on a blockchain (*i.e.* a distributed ledger)[3]. In such a deployment, all transactions involving the ICAs (*i.e.* messages to or from each credentialing ICA) will be recorded on the blockchain, and so it will be obvious (*i.e.* publicly visible) if an ICA uses a credential for any purpose (for example, sending it to a pseudonymizing ICA to be pseudonymized, or sending it to a PKG to obtain a private key). Thus, this impersonation attack will have essentially no chance of success because the nefarious actions of the credentialing ICA will immediately be detected by others.

Alternatively, if there is no smart contract implementation, it is still possible to eliminate the risk that an HbC credentialing ICA will impersonate Alice. To achieve this, we simply stipulate that a valid credential must be digitally signed by more than one credentialing ICA (see Section 4.6 for details on how this can be done). Since the credentialing ICA in this attack is honest-but-curious, it is not possible for it to collude with other credentialing ICAs to get them to sign this impersonation credential.

4.6. Security against Malicious Entities

Malicious entities break whichever rules they wish if this will enable them to achieve some nefarious purpose. They may create invalid data structures and non-standard protocol messages, and may collude with other malicious entities at will. We begin by looking at malicious entities acting alone and then explore various collusions that may occur.

A Malicious Entity Acting Alone

Malicious User. **A malicious user Bob (working alone) wishes to learn Alice's private key so that he can decrypt ciphertext that was encrypted for her.** As with the HbC Bob, a malicious Bob could try to convince a credentialing ICA to create for him a credential for Alice's identity. A sufficiently trustworthy proof of identity ownership mechanism (outside the scope of IBE algorithms and protocols) will preclude this attack.

The other possibility is that a malicious Bob might create a credential for Alice's identity by himself (*i.e.* without interacting with a credentialing ICA) and then use this forged credential in subsequent interactions with pseudonymizing ICAs or with the PKG. This attack is precluded because Brands has proved ([7], pp. 149-154) that his digital credentials are unforgeable in the random oracle model under the strongest attack model (*i.e.* the attacker can engage in polynomially many executions of the issuing protocol, can arbitrarily interleave protocol executions, and can select an arbitrary attribute tuple at the start of each new protocol execution).

Therefore, a malicious user Bob (acting alone) can be prevented from learning Alice's private key.

Malicious PKG. **A malicious private key generator (working alone) wishes**

[3]Ibid.

to compute Alice's private key so that it can decrypt ciphertext that was encrypted for her. The malicious PKG working alone has no more data to manipulate or process than an HbC PKG. Consequently, like the HbC PKG, security against a malicious PKG is provided by the elliptic curve discrete logarithm problem, and by having an open-world setting with a sufficiently large identity space.

Therefore, a malicious PKG (acting alone) can be prevented from learning Alice's private key.

Malicious pseudonymizing ICA. A malicious pseudonymizing intermediate CA (acting alone) wishes to learn Alice's private key so that it can decrypt ciphertext that was encrypted for her. The malicious pseudonymizing ICA working alone has no more data to manipulate or process than an HbC pseudonymizing ICA. Consequently, the elliptic curve discrete log problem and an open-world setting with a sufficiently large identity space can preclude impersonation attacks.

Therefore, a malicious pseudonymizing ICA (acting alone) can be prevented from learning Alice's private key.

Malicious credentialing ICA. A malicious credentialing intermediate CA (acting alone) wishes to learn Alice's private key so that it can decrypt ciphertext that was encrypted for her. As with the HbC credentialing ICA, the malicious credentialing ICA working alone is unable to undetectably impersonate Alice if it is implemented as a smart contract on a blockchain. Alternatively, if there is no smart contract implementation, we can stipulate that a valid credential must be digitally signed by more than one credentialing ICA; since in this scenario we are modeling a malicious credentialing ICA that acts alone, this attack is precluded.

Therefore, a malicious credentialing ICA (acting alone) can be prevented from learning Alice's private key.

Collusions among Malicious Entities

The construction described in Section 3.2 has four types of entities and multiple instances of each type (other than the PKG) in a full deployment. Thus, we need to consider the implications on the security of ciphertext intended for Alice if there are collusions among 2, 3, or all 4 types of entities.

User-PKG. If a malicious user Bob colludes with a malicious PKG, then Bob can reveal Alice's identity (*i.e.* her e-mail address) to the PKG, and the PKG can compute and return Alice's private key to Bob. An open-world setting cannot preclude this attack because we can assume that Bob knows a specific victim (*i.e.* Alice) that he wishes to target. Thus, this attack cannot be prevented.

Since in a deployment of any reasonable size, it is impossible to guarantee that every user is honest (or at least honest-but-curious), the only protection against this form of collusion is to strive to make the PKG at least honest-but-curious. Implementing the PKG as a smart contract is one possible way to achieve this.

User-Pseudonymizing ICA, or User-Credentialing ICA. If a malicious user Bob colludes with either a malicious pseudonymizing ICA or a malicious cre-

dentialing ICA, then the ICA can create a credential that Bob can use to impersonate Alice with the PKG. As with the Bob-PKG collusion, an open-world setting cannot preclude this (because we can assume that Bob knows the e-mail address of his victim Alice). However, we can reduce the probability of a successful impersonation to an arbitrarily small value by requiring multiple signatures for a valid credential and for a valid pseudonym, and by using a pseudorandom algorithm to determine who the signers must be.

Let there be v ICAs in the environment (we can use this mechanism identically with pseudonymizing ICAs and with credentialing ICAs, and so we will use the terms "ICA" and "credential" here for simplicity) and let these ICAs be uniquely labeled with the identifiers "1" through "v". As a concrete example, assume that a valid credential requires 3 signatures. The entity that desires the credential chooses an ICA (say ICA_i), and uses Brands' issuing protocol to interact with ICA_i to create h and its digital signature (c_{0i}', r_{0i}'). (Note that the attribute a_2 in h is the value i, the identifier of the original signer of the credential, ICA_i.) The first $\log_2 v$ bits of $(c_{0i}' \cdot r_{0i}' \bmod q)$ that are different from i determine the second signer. Say these bits are the value j. The entity interacts with ICA_j to obtain a second signature on h, (c_{0j}', r_{0j}'). The first $\log_2 v$ bits of $(c_{0j}' \cdot r_{0j}' \bmod q)$ that are different from i and j determine the third signer. Say these bits are the value k. The entity interacts with ICA_k to obtain a third signature on h, (c_{0k}', r_{0k}'). The set of values $\{h, \text{signature}_i, \text{signature}_j, \text{signature}_k\}$ forms a valid credential.

In the case of a malicious ICA, the ICA would clearly choose itself (or another known malicious ICA) as the first (original) signer, but would have no control over the choice of second and third signers. Therefore, if at least one of these other signers is honest or honest-but-curious, a malicious pseudonymizing ICA will be unable to create a valid pseudonym for successful impersonation, and a malicious credentialing ICA will be unable to create a valid credential for successful impersonation (unless the credentialing ICA can convincingly prove ownership of Alice's identity to these other signers).

The probability of successful impersonation becomes successively smaller as the required number of signatures for a valid credential is increased. For any given deployment, the risk can be set at an arbitrarily low level by appropriately setting the number of required signatures (however, the tradeoff is the increased computation required for creating and verifying the full set of credential signatures).

PKG-Pseudonymizing ICA, or PKG-Credentialing ICA. If a malicious PKG colludes with either a malicious pseudonymizing ICA or a malicious credentialing ICA, then requiring additional signatures to create a valid credential will not prevent an attack (because the malicious PKG will not check the validity of the credential in any case). However, an open-world setting with a sufficiently large identity space will reduce the risk of an attack resulting from collusion between a credentialing ICA and the PKG because the credentialing ICA and the PKG will not trivially know Alice's e-mail address. Furthermore, if the honest (or honest-but-curious) user has used a chain of pseudonymizations, then the

PKG would need to collude with every pseudonymizing ICA in the chain; this can only be successful if every pseudonymizing ICA in the chain is malicious (a risk that can be made arbitrarily small by choosing a sufficiently long chain).

Credentialing ICA-Pseudonymizing ICA. If a malicious credentialing ICA and a malicious pseudonymizing ICA collude, they may create an invalid credential and an invalid pseudonym, but will need to create at least one valid pseudonym to send to the honest (or honest-but-curious) PKG. Thus, a requirement for multiple signatures on the pseudonym will reduce the risk of impersonation to an arbitrarily low level. Furthermore, in an open-world setting with a sufficiently large identity space, the ICAs will not trivially know Alice's e-mail address, which also reduces the risk of an impersonation attack.

Credentialing ICA-Pseudonymizing ICA-PKG. If a malicious credentialing ICA and a malicious pseudonymizing ICA collude with a malicious PKG, then an invalid credential and an invalid pseudonym can be created and these will be accepted by the PKG. Note that the open-world setting will not reduce the risk of an attack in this scenario: when the honest (or honest-but-curious) user Alice contacts the credentialing ICA to obtain her credential, the credentialing ICA will learn Alice's e-mail address and subsequently collude with the pseudonymizing ICA and the PKG to learn her private key.

User-Credentialing ICA-Pseudonymizing ICA. If a malicious user Bob colludes with both a malicious credentialing ICA and a malicious pseudonymizing ICA, then Bob can supply the target identity (*i.e.* Alice's e-mail address) and the ICAs can create an invalid credential and an invalid pseudonym. However, the ICAs will need to create at least one valid pseudonym to send to the honest (or honest-but-curious) PKG, and so the requirement for multiple signatures on the pseudonym will reduce the risk of impersonation to an arbitrarily low level.

User-Pseudonymizing ICA-PKG, or User-Credentialing ICA-PKG, or User-Credentialing ICA-Pseudonymizing ICA-PKG. If a malicious user Bob can collude with a malicious PKG, then it does not matter whether a malicious credentialing ICA and/or a malicious pseudonymizing ICA is also available in the environment. As mentioned above, this attack cannot be prevented. Thus, the only mitigation is to strive to make the PKG at least honest-but-curious (perhaps by implementing it as a smart contract).

5. Discussion

As shown in Section 4, our proposed construction has multiple types of entities, and a threat model for this environment can consider any of them to be honest-but-curious or malicious (instead of purely honest) and can consider collusions among various collections of malicious attackers. The security analysis reveals that

- Some essential requirements for security are beyond the scope of IBE algorithms and protocols (such as the existence of a trustworthy mechanism to prove ownership of an e-mail identity, and the use of an open-world setting

with a sufficiently large identity space), and

- The vast majority of the attacks in the threat model can be prevented by ensuring that the requirements in the previous bullet are satisfied, along with appropriately choosing elliptic curve parameters, requiring a sufficiently long chain of pseudonymizing ICAs, and requiring sufficiently many signatures on a valid pseudonym and credential.

Notably, the analysis also reveals that

- Some attacks cannot be prevented (such as a malicious PKG that learns Alice's e-mail address, for example through collusion with a malicious user Bob or through collusion with a malicious credentialing ICA that was contacted by Alice). However, even these attacks can potentially be mitigated by a smart contract implementation of some components in the system.

6. Conclusions

This paper provides an extensive security analysis of a proposed construction to reduce the trust in the private key generator (PKG) of an identity-based encryption (IBE) system. In particular, the analysis includes all the various types of entities in the environment, models the attacking entities as either *honest-but-curious* or *malicious*, and considers collusions among all collections of malicious entities.

The security analysis shows that the vast majority of attacks to learn a user's plaintext can be prevented by choosing appropriate system parameters, employing security mechanisms outside the scope of IBE (such as identity authentication and the use of an open-world setting), and requiring the participation of multiple randomly-chosen entities in specific tasks. On the other hand, the few attacks that cannot be prevented can nevertheless be mitigated through other means, such as the use of smart contract implementations in a given deployment.

Our hope is that this proposed construction and its security analysis will generate renewed interest in the study and use of IBE for real-world environments.

Acknowledgements

This work was partially supported by the *Natural Sciences and Engineering Research Council of Canada* (NSERC).

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

[1] Shamir, A. (1985) Identity-Based Cryptosystems and Signature Schemes. *Advances in Cryptology—Proceedings of CRYPTO*'84, Vol. 196, Santa Barbara, 19-22 August 1984, 47-53.

[2] Boneh, D. and Franklin, M. (2001) Identity-Based Encryption from the Weil Pairing

(Extended Abstract). *Advances in Cryptology—Proceedings of Crypto* '2001, Vol. 2139, Santa Barbara, 19-23 August 2001, 231-229. http://eprint.iacr.org/2001/090/ https://doi.org/10.1007/3-540-44647-8_13

[3] Boneh, D. and Franklin, M. (2003) Identity-Based Encryption from the Weil Pairing. *SIAM Journal on Computing*, **32**, 586-615. https://doi.org/10.1137/S0097539701398521

[4] Chow, S.S.M. (2009) Removing Escrow from Identity-Based Encryption. *Public Key Cryptography—PKC* 2009, Vol. 5443, Irvine, 18-20 March 2009, 256-276. https://doi.org/10.1007/978-3-642-00468-1_15

[5] Bendlin, R., Krehbiel, S. and Peikert, C. (2013) How to Share a Lattice Trapdoor: Threshold Protocols for Signatures and (H)IBE. *ACNS* 2013: *Applied Cryptography and Network Security*, Vol. 7954, Banff, 25-28 June 2013, 218-236. https://doi.org/10.1007/978-3-642-38980-1_14

[6] Emura, K., Katsumata, S. and Watanabe, Y. (2022) Identity-Based Encryption with Security against the KGC: A Formal Model and its Instantiations. *Theoretical Computer Science*, **900**, 97-119. https://doi.org/10.1016/j.tcs.2021.11.021 https://www.sciencedirect.com/science/article/pii/S030439752100699X

[7] Brands, S. (2000) Rethinking Public Key Infrastructure and Digital Certificates: Building in Privacy. The MIT Press, Cambridge, MA. https://doi.org/10.7551/mitpress/5931.001.0001

[8] Brands, S. (2002) A Technical Overview of Digital Credentials. Credentica. Credentica Technical Paper. http://www.credentica.com/overview.pdf

[9] Koblitz, N. (1987) Elliptic Curve Cryptosystems. *Mathematics of Computation*, **48**, 203-209. https://doi.org/10.1090/S0025-5718-1987-0866109-5

[10] Miller, V. (1986) Use of Elliptic Curves in Cryptography. *Advances in Cryptology—Proceedings of CRYPTO* '85, Vol. 218, Santa Barbara, 18-22 August 1985, 417-426. https://doi.org/10.1007/3-540-39799-X_31

[11] Galbraith, S., Harrison, K. and Soldera, D. (2002) Implementing the Tate Pairing. *Algorithmic Number Theory Symposium*: 5*th International Symposium* (*ANTS-V*), Vol. 2369, Sydney, 7-12 July 2002, 324-337. https://doi.org/10.1007/3-540-45455-1_26

[12] Miller, V. (2004) The Weil Pairing, and its Efficient Calculation. *Journal of Cryptology*, **17**, 235-261. https://doi.org/10.1007/s00145-004-0315-8

[13] Adams, C. and Lloyd, S. (2003) Understanding PKI: Concepts, Standards, and Deployment Considerations. 2nd Edition, Addison-Wesley, Boston.

[14] Housley, R. and Polk, T. (2001) Planning for PKI: Best Practices Guide for Deploying Public Key Infrastructure. Wiley, New York.

[15] Szabo, N. (1996) Smart Contracts: Building Blocks for Digital Markets. https://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart_contracts_2.html

[16] Levi, S.D. and Lipton, A.B. (2018) An Introduction to Smart Contracts and Their Potential and Inherent Limitations. Harvard Law School Forum on Corporate Governance. https://corpgov.law.harvard.edu/2018/05/26/an-introduction-to-smart-contracts-and-their-potential-and-inherent-limitations/