cryptography

MDPI

Article

Improving User Privacy in Identity-Based Encryption Environments

Carlisle Adams

School of Electrical Engineering and Comuter Science (EECS), University of Ottawa, Ottawa, ON M3J 1P3, Canada; cadams@uottawa.ca

Abstract: The promise of identity-based systems is that they maintain the functionality of public key cryptography while eliminating the need for public key certificates. The first efficient identity-based encryption (IBE) scheme was proposed by Boneh and Franklin in 2001; variations have been proposed by many researchers since then. However, a common drawback is the requirement for a private key generator (PKG) that uses its own master private key to compute private keys for end users. Thus, the PKG can potentially decrypt all ciphertext in the environment (regardless of who the intended recipient is), which can have undesirable privacy implications. This has led to limited adoption and deployment of IBE technology. There have been numerous proposals to address this situation (which are often characterized as methods to reduce trust in the PKG). These typically involve threshold mechanisms or separation-of-duty architectures, but unfortunately often rely on non-collusion assumptions that cannot be guaranteed in real-world settings. This paper proposes a separation architecture that instantiates several intermediate CAs (ICAs), rather than one (as in previous work). We employ digital credentials (containing a specially-designed attribute based on bilinear maps) as the blind tokens issued by the ICAs, which allows a user to easily obtain multiple layers of pseudonymization prior to interacting with the PKG. As a result, our proposed architecture does not rely on unrealistic non-collusion assumptions and allows a user to reduce the probability of a privacy breach to an arbitrarily small value.

Keywords: identity-based encryption; reducing trust; privacy; digital credentials

Citation: Adams, C. Improving User Privacy in Identity-Based Encryption Environments. *Cryptography* **2022**, *6*, 55. https://doi.org/10.3390/cryptography6040055

Academic Editor: Josef Pieprzyk

Received: 8 August 2022
Accepted: 4 November 2022
Published: 9 November 2022

1. Introduction

At the Crypto conference in 1984, Adi Shamir proposed the concept of identity-based cryptography [1], in which a user's identity can effectively be used as a public key for cryptographic purposes (e.g., for encryption, or for verifying a digital signature). Such a scheme, if realizable, would eliminate the need for public key certificates in a given environment, removing the complexity of creating and managing every certificate for the lifetime of its embedded public key, as well as the complexity of a user Alice validating a (potentially long) chain of certificates in order to trust the public key of another user Bob.

In 2001, Boneh and Franklin proposed the first construction for identity-based encryption (IBE) that efficiently realized Shamir's concept [2] (see also [3]). This construction is elegant as well as efficient, but includes a central system authority, the Private Key Generator (PKG), that computes user private keys upon request (using its own master private key). This is acceptable if the PKG is fully trusted by all parties in the environment. However, there may be specific environments, or specific situations within a given environment, in which a user would desire privacy from the PKG (in particular, Alice may be sent some ciphertext and she would like to be the *only* entity able to decrypt it).

Schemes for improving user privacy in IBE (equivalently, reducing trust in the IBE PKG) have been proposed by a number of researchers; see, for example [2–6]. However, these previous approaches have relied on assumptions that do not match real-world environments: specifically, they assume guaranteed perpetual non-collusion between specific entities in the construction—even entities within a single company—which is difficult (often impossible) to achieve in real deployments.

In this paper, we propose a scheme for improving user privacy that does not rely on unrealistic non-collusion assumptions (therefore, it is more deployable in real environments) and that allows a user Alice to reduce the probability that some other entity will learn her private key to an arbitrarily low level (so that it also has higher security in real environments). Thus, Alice can enjoy the benefits of IBE without fear that someone else will decrypt her ciphertext. Our construction differs from previous work in several non-trivial ways, including that a digital credential is used as the blind token issued by an ICA, and that one of the credential attributes is computed as the bilinear map of a given elliptic curve point. As shown in Section 3 below, these novel features provide the foundation that enables a significantly higher degree of protection of Alice's privacy than previous work. The scientific contribution of our research is to construct an IBE scheme with greater privacy and deployability than prior proposals.

The remainder of the paper is organized as follows: Section 2 provides a brief background on the mathematical tools needed to understand our proposal; Section 3 outlines some of the prior approaches to improving privacy; Section 4 presents our proposed constructions; Section 5 describes some implementation aspects, summarizes the results of our security analysis, and discusses limitations of our proposal; Section 6 concludes the paper.

2. Materials and Methods

This section provides an introduction to the technologies needed for understanding the remainder of this paper: digital credentials; elliptic curves; bilinear maps; and identity-based encryption. Our proposed solution makes use of all these technologies to improve user privacy in IBE deployments (Note that the availability of an anonymous communications channel (such as Tor [7]) is assumed in our proposed environment).

Prior work has not eliminated the need for users to fully trust the authorities in the architecture. There have been several proposals, but these typically mandate that certain entities can never collude, or that no more than $\tau - 1$ entities can collude, which can be impossible to guarantee in real-world environments (therefore, some level of trust is always needed). Our work uses the technologies described in this section to construct an architecture in which the risk of privacy breach due to collusion can be reduced to an arbitrarily small level because collusion would have to occur among η randomly-selected entities, where the user chooses the value η. More detail is given in Section 3 below.

2.1. Digital Credentials

A *digital credential*, as proposed by Brands [8,9], is a public key (and corresponding authority signature) which enables a specific kind of privacy in network interactions. Embedded in Alice's public key are attributes about her (such as *name*, *home address*, *job title*, and so on) which she can choose to reveal, or to keep unconditionally hidden, in any given online transaction. Her digital credential allows her to remain anonymous while proving that she validly possesses an attribute that is required for access to a web site, for example.

A digital credential implementation consists of a pair of protocols: an *issuing protocol* in which Alice interacts with a CA ("Certification Authority", sometimes referred to as a "Credential Authority" in this context) to obtain the CA's digital signature on her public key; and a *showing protocol* in which Alice interacts with a verifier, Bob, to reveal selected attributes in her public key.

The system parameters known to all participants are p (a prime of sufficient size for security, such as 2048 bits or more), q (a prime that divides $(p - 1)$, g_0 (a generator of the q-order subgroup of $\mathbb{Z}_p{}^*$), and H() (a cryptographically strong one-way hash function that produces outputs in $\mathbb{Z}_q$). The CA chooses $m + 1$ random values $y_0, y_1, \ldots, y_m$ in $\mathbb{Z}_q{}^*$ and computes $h_0 = g_0{}^{y_0}$, $g_1 = g_0{}^{y_1}$, $\ldots$, $g_m = g_0{}^{y_m}$ (all exponentiations done modulo p). The public key of the CA is the set of integers $\{h_0, g_0, g_1, \ldots, g_m\}$ and its corresponding private key is the set of integers $\{y_0, y_1, \ldots, y_m\}$.

Assume that Alice has m attributes, $\{x_1, \ldots, x_m\}$, all in $\mathbb{Z}_q$. Alice chooses a random value $\alpha \in \mathbb{Z}_q$ (which she does not share with anyone) and constructs her public key $h = ({g_1}^{x_1} \cdot {g_2}^{x_2} \cdot \ldots \cdot {g_m}^{x_m} \cdot h_0)^{\alpha} \; mod \; p$. Her private key is the set of values $\{x_1, \ldots, x_m, \alpha\}$.

In the issuing protocol, Alice obtains from the CA a digital signature on h. This (blinded) signature has the form $({c_0}', {r_0}')$ and anyone can verify this signature by checking that ${c_0}' = H(h, \; {g_0}^{{c_0}'} \cdot h^{{r_0}'} \; mod \; p)$ (The reader is referred to [8,9] to see the details of the issuing protocol). If the signature verifies using this equation, the verifier is convinced that h is a valid public key and that whoever knows the corresponding private key is the legitimate owner of the attributes contained in this public key.

In the showing protocol, Alice can reveal one or more of her attribute values x_i, while keeping the remaining attribute values (as well as the value α) unconditionally hidden. This is done using a *zero-knowledge proof-of-knowledge* technique (Again, the reader is referred to [8,9] to see the details of the showing protocol).

2.2. Elliptic Curves

An elliptic curve is defined by the general equation $y^2 + \beta xy + \gamma y = x^3 + \omega x^2 + \varepsilon x + \zeta$, where x and y are variables and $\beta, \gamma, \omega, \varepsilon$, and ζ are coefficients. When used for cryptographic purposes, the variables and coefficients are elements of a finite field ($GF(\rho)$ for ρ prime, or $GF(2^{\ell})$ for ℓ an integer greater than 1), and a simpler form of the curve equation (which sets some of the coefficient values to zero or one) is used: $y^2 = x^3 + ax + b$ (for $GF(\rho)$) or $y^2 + xy = x^3 + ax^2 + b$ (for $GF(2^{\ell})$). These curves are typically denoted $E_{\rho}(a, b)$ or $E_{2^{\ell}}(a, b)$.

For a given finite field and a given choice of coefficients a and b, each pair of values x and y that satisfy the curve equation represents a point on the curve. The complete set of points, along with a special point at infinity and a specific operation for addition, creates a finite additive group (over which cryptographic operations can be computed). The use of elliptic curves for cryptography was first proposed by Koblitz [10] and Miller [11].

For carefully-chosen curves and parameters, Elliptic Curve Cryptography (ECC, including ECDSA and ECDH, for example) is believed to provide high levels of security with much smaller keys than are required for comparable cryptographic operations performed over multiplicative groups (DSA and DH, for example). Elliptic curve groups are also used as the basis for security in some implementations of Identity Based Encryption (IBE).

2.3. Bilinear Maps

A bilinear map can be instantiated as a function $\hat{e} : G_1 \times G_1 \rightarrow G_T$, where G_1 is an elliptic curve additive group and G_T is a multiplicative group (often a finite field over the integers). G_1 and G_T have prime order q and $\hat{e}$ has the following properties.

- Bilinear: $\forall P_1, P_2 \in G_1$ and $\forall a, b \in {\mathbb{Z}_q}^*$, $\hat{e}(aP_1, bP_2) = \hat{e}(P_1, P_2)^{ab}$
- Non-degenerate: $\forall$ non-trivial points $P_1 \in G_1$, $\hat{e}(P_1, P_1) \neq 1$
- Computable: $\forall P_1, P_2 \in G_1, \hat{e}(P_1, P_2)$ is efficiently computable.

Bilinear maps (particularly when used for cryptographic purposes) are typically implemented using either the Tate pairing [12] or the Weil pairing [13]. Note that maps instantiated as $\hat{e} : G_1 \times G_2 \rightarrow G_T$ also exist and are preferred in many environments for their efficiency and security; see, for example [14].

2.4. Identity-Based Encryption

Public key cryptography typically requires a binding between a public key (which looks like a very large random integer) and a user's identity. This binding is needed so that if Bob wishes to encrypt some data for Alice, he knows which public key to use for her. In Public Key Infrastructure (PKI) environments, the binding is provided by a public key certificate (a data structure containing at least Alice's identity (such as her name or e-mail address) and Alice's public key, digitally signed by a Certification Authority (CA) whose public verification key Bob is able to trust; see, for example [15,16]).

Certificates work well and are widely used, but have a number of limitations and implementation difficulties (including distribution, validation, renewal, and revocation) that can be significant drawbacks in large-scale deployments. To mitigate these problems, Shamir proposed [1] the concept of *identity based cryptography, IBC* (including *identity based encryption (IBE)* and *identity based signatures (IBS)*), in which Alice's identity is her public key, so that anyone who knows Alice's identity can immediately encrypt for her or veryify her digital signature (without the need for a public key certificate). In 2001, Boneh and Franklin were the first researchers to find an efficient way to realize Shamir's IBE concept [2,3]; they proposed the use of a bilinear map (also known as a *bilinear pairing*) over elliptic curve groups. Since that time, a number of other researchers have proposed implementations of IBE and IBS using other mathematical primitives such as matrices (e.g., [17]) and lattices (e.g., [6,18]).

An Example IBE Instantiation

The following construction (proposed in [2,3]) serves to illustrate one example of how an IBE scheme can be instantiated. Let G be a generator of the group G_1, and let PKG (the *Private Key Generator*, referred to in some papers as the *Key Generation Center*, KGC) be a trusted authority with master private key $t \in \mathbb{Z}_q{}^*$ and corresponding master public key $T = tG$. Along with T, the system parameters (known by all participants in the environment) include the following functions:

- $H_1 : \{0,1\}^* \rightarrow G_1$
- $H_2 : \{0,1\}^n \times \{0,1\}^n \rightarrow \mathbb{Z}_q{}^*$
- $H_3 : G_T \rightarrow \{0,1\}^n$
- $H_4 : \{0,1\}^n \rightarrow \{0,1\}^n$

Let user Alice have identity *"alice@gmail.com"*. Her public key (computable by anyone that knows her identity) is $I_A = H_1(\text{"}alice@gmail.com\text{"})$. Note that I_A is an elliptic curve point in G_1. In the original Boneh and Franklin scheme, to encrypt message $msg \in \{0,1\}^n$ for Alice, Bob will do the following five steps.

1. Compute $\mu = \hat{e}(I_A, T)$
2. Choose $\sigma \in_R \{0,1\}^n$
3. Compute $r = H_2(\sigma || msg)$
4. Compute $z = \mu^r$
5. Compute ciphertext $c = (u, v, w) = (rG,\ \sigma \oplus H_3(z),\ msg \oplus H_4(\sigma))$

To decrypt the ciphertext (u, v, w), Alice will do the following four steps.

1. Obtain her private key from the PKG (To do this, she will prove to the PKG that she validly owns the e-mail address *"alice@gmail.com"* and the PKG will compute Alice's identity I_A from this address, compute $K_A = tI_A$, and give K_A to Alice as her private key).
2. Compute $\hat{e}(K_A, u)$ (Note that $\hat{e}(K_A, u) = \hat{e}(tI_A, rG) = \hat{e}(rI_A, tG) = \hat{e}(rI_A, T) = \hat{e}(I_A, T)^r = \mu^r = z$).
3. Compute $\sigma = v \oplus H_3(z),\ msg = w \oplus H_4(\sigma),\ r = H_2(\sigma || msg)$
4. If $u = rG$, return msg. Otherwise, return *invalid*.

At a high level, the decryption logic is as follows: given the ciphertext (u, v, w), Alice uses u to compute z, then uses z and v to compute σ, uses σ and w to compute msg, and uses msg and σ to compute r. If rG is equal to the originally-received u, then the ciphertext is deemed to be valid and msg is the correct plaintext.

3. Results

The construction above elegantly achieves Shamir's vision of a user identity serving as a public key so that no certificate is needed. Alice can obtain her private key from the PKG at any time she wishes (including after Bob has encrypted a message for her, which is

different from traditional public key algorithms that create the key pair prior to first use of the public portion).

3.1. The Private Key Generator (PKG)

Because Alice is not able to compute her own private key, this means that the PKG must be completely trusted. In fact, the PKG must be trusted by all users in the environment, since it generates every user's private key. The PKG can therefore decrypt all ciphertext, regardless of who the recipient is intended to be. This is sometimes referred to as the "key escrow problem" of IBE, but "escrow" typically implies an authority holding an item for safekeeping upon request, whereas the IBE architecture goes beyond this: all private keys are known to the PKG, whether the users desire this or not.

Note that unlike the situation for digital signatures, it is well known that there can be tremendous value in holding backup copies of private decryption keys in a government or commercial setting (see, for example, pp. 97–98 of [15]). Business continuity demands that critical data can still be accessed by someone in the organization if the person whose key was used for the encryption leaves or becomes incapacitated in some way. IBE, therefore, has an inherent decryption key backup characteristic that can be very beneficial in such environments. However, again, IBE goes well beyond a backup feature to ensure that critical documents are not lost; rather, every piece of ciphertext (no matter how trivial or personal) can potentially be decrypted and read by the PKG. This puts the privacy of all users at risk.

3.2. Threat Model and Security Goal

The threat model defined for this environment is that the PKG is less than completely *honest*. In particular, if the PKG is *honest-but-curious*, it will execute all required IBE protocols and computations correctly but, in addition, will use whatever means it has at its disposal to try to learn Alice's private key. In a more extreme setting, the PKG may be *malicious*, in which case it cannot be depended upon to execute the IBE protocols and computations correctly, and may collude with other malicious entities to try to learn Alice's key.

For both PKG characterizations (i.e., *honest-but-curious* and *malicious*), the security goal is to *keep Alice's private key hidden from all entities in the environment (other than Alice)*. It is possible that some *malicious* entities may cause denial-of-service (so that Alice does not obtain her private key), but the goal of our construction is to ensure that no entities will succeed in compromising the confidentiality of Alice's key.

3.3. Previous Proposals to Reduce Trust in the PKG

The trust required in the PKG, and the privacy implications if the PKG is not completely *honest*, has potentially hurt the widespread adoption of IBE, particularly in public (i.e., non-business) settings. Consequently, many researchers (including Boneh and Franklin with their original scheme) have proposed mechanisms to eliminate, or at least mitigate, the "escrow problem". This is often characterized as techniques to reduce the trust in the PKG, but can equally be characterized as techniques to improve user privacy in IBE environments.

The numerous techniques that have been proposed vary significantly in their details, but most fall into one of two main categories: threshold techniques and separation techniques.

3.3.1. Threshold Techniques

In a threshold scheme, there are multiple PKGs and a subset of these is required in order to compute the private key for a user. In particular, in an (n, τ) threshold scheme, any collection of τ or more PKGs (out of a total of n PKGs) can compute the user private key. Each PKG has only a piece (a *share*) of the master private key; any τ or more of these shares will reconstruct the master private key so that the user private key can be computed. Of course, it is important to ensure that no PKG learns the reconstructed master private key at the end of the protocol (otherwise that PKG would then need to be fully trusted). Furthermore, it is also necessary that no single entity begins with the master private key

and then splits it into shares to give to the n PKGs (otherwise this entity would then need to be fully trusted).

Threshold techniques for IBC environments have been explored in [2,5], for example. Boneh and Franklin [2] use standard techniques such as those described in [19], whereas Bendlin et al. [5] use threshold protocols that are specially developed for lattice-based cryptography and are applicable to both traditional public-key cryptography and identity-based cryptography settings.

Multi-Party Computation (MPC) schemes, in which each of several participants contributes to the computation of a function output but no participant learns the private input data of any other participant (see [20]), can be helpful in realizing effective threshold schemes. MPC is flexible in that τ can be chosen to be any value between 1 and n, and these schemes can be designed to work correctly even when there are subsets of malicious or non-responsive participants.

With threshold techniques in general, the requirement that no entity ever learns the master private key (even prior to the distribution of *shares*) may be non-trivial to achieve in practice, but it can be done using (for example) the method developed by Gennaro et al. in [21]—this method ensures that master private key shares are generated directly on the distributed PKGs so that they are never collected in a single location. On the other hand, instantiating multiple PKGs and absolutely guaranteeing (beyond any doubt) that no more than $\tau - 1$ malicious PKGs ever collude would clearly be difficult to accomplish in many real-world environments.

3.3.2. Separation Techniques

In the original IBE schemes, the PKG implicitly performs two tasks: it verifies the identity of the requesting user; and it then computes the private key that corresponds to the verified identity. In separation techniques, these two tasks are explicitly divided and given to two independent entities. Thus, there is a PKG that computes private keys, but there is also an *Intermediate Certification Authority* (Intermediate CA, or ICA, sometimes referred to as an *Identifying CA*) that verifies the identity. The ICA issues a certificate (in the form of a blinded token that is signed by the ICA) to the user; the user subsequently presents this certificate to the PKG in order to obtain the user's private key.

The recent paper by Emura et al. [6] provides a formalization of an IBE scheme that is based on the separation architecture proposed by Chow [4]. Emura et al. define three security notions—describing *indistinguishability and anonymity against chosen plaintext attacks* for the user, for the ICA, and for the PKG—and give two instantiations of their proposal (one on lattices and one on bilinear maps (pairings)) that they prove to be secure relative to these defined notions. A significant advance over Chow's work is that in the Emura et al. scheme the ICA is not assumed to be fully trusted; therefore this proposal provides protection against the key escrow problem in a more realistic setting. Another advance is that the lattice instantiation creates the first post-quantum IBE to address the key escrow problem based on Chow's work. However, an important limitation with the pairing-based instantiation of Emura et al. is that the domain of the pairing function must be a multiplicative group (because the blinded token in the ICA-issued certificate is specified to be a product of group elements) and so this construction cannot be implemented over an elliptic curve group (Unfortunately, working over the additive group of elliptic curve points is where IBE gets most of its computational and bandwidth efficiency for high security levels).

With a separated architecture, the PKG computes a private key but does not know which user this key is for. The PKG could try decrypting all ciphertext with this key in an attempt to find some readable plaintext; however, not only would this be an arduous task (especially in any large organization), but it would not work in any case since the computed key is a blinded (i.e., randomized) version of the true private key. On the other hand, if the PKG and the ICA collude, all privacy is immediately lost (which is why in the Emura et al. scheme, the security proof for the ICA assumes that the ICA has no access to the PKG). As

mentioned in the previous subsection on threshold techniques, ensuring continuous and perpetual non-collusion is difficult in any real-world environment, but it may be especially hard if the ICA and the PKG are personnel in the same company.

Combining separation with thresholding techniques (at the PKG, or at the ICA, or at both) may hold some promise, but such approaches quickly get very complex and may still not completely eliminate the escrow problem in practice because of the required non-collusion assumption.

4. Our Proposal

As with previous work (such as [4,6]), we employ a separation architecture and introduce an Intermediate Certification Authority (ICA) that creates a certificate for Alice's identity. However, unlike this previous work, the ICA in our scheme produces a Brands' *digital credential* as the certificate. Thus, Alice interacts with the ICA, proving ownership of her identity in order to obtain a digital credential, and she subsequently interacts with the PKG using this credential in order to obtain her IBE private key.

In the *Basic Scheme* presented below, there is a single ICA in the construction, which is identical to previous separation proposals [4,6]. However, the subsequent *Augmented Scheme* instantiates multiple ICAs; this enhancement allows us to address the possibility of collusion between the PKG and the ICA, delivering a level of privacy that the previous proposals cannot provide.

4.1. Basic Scheme

Let q be a prime and let $E_\rho(a, b)$ be an elliptic curve with generator point G. Let $\hat{e}$ be a bilinear pairing $G_1 \times G_1 \rightarrow G_T$, where G_1 is the group generated by G, G_T is a multiplicative cyclic group, and G_1 and G_T are both of order q (We are using the *symmetric* pairing $G_1 \times G_1 \rightarrow G_T$, rather than the *asymmetric* pairing $G_1 \times G_2 \rightarrow G_T$, in order to build directly on the original IBE scheme by Boneh and Franklin. However, straightforward modifications allow our proposed constructions to accommodate the asymmetric pairing, if desired).

Alice contacts the ICA and proves that her identity (i.e., her e-mail address) is, say, "alice@gmail.com". The ICA computes $P = H_1(\text{"}alice@gmail.com\text{"})$, where P is a point on the curve $E_\rho(a, b)$, and computes $\xi = \hat{e}(P, P)$. Using Brands' *issuing protocol*, Alice constructs the digital credential public key $h = (g_1{}^{a_1} \cdot g_2{}^{a_2} \cdot h_0)^\alpha \ mod\ p$, and the ICA (blindly) creates its corresponding signature $(c'_0,\ r'_0)$, where the first attribute a_1 in h is the value ξ and the second attribute a_2 is set to be the unique identifier for the ICA.

When Alice interacts with the PKG, she presents h, the signature on h, and the point P. With Brands' *showing protocol* she reveals a_1 and a_2. The PKG verifies the signature, confirms that a_2 is the identifier for the ICA whose public key verifies the signature, and confirms that $\hat{e}(P, P) = a_1$ (If the ICA is acting correctly (i.e., if the ICA is *honest* or *honest-but-curious*), then because a_1 is an attribute in the credential, the PKG is convinced that P is validly associated with Alice; in particular, P must be the hash of Alice's e-mail address). The PKG therefore computes the IBE private key $K_A = tP$ and gives K_A to Alice. This scheme is shown in Figure 1.

The above process successfully splits the original IBE PKG into two pieces, an ICA and a PKG (as was proposed in previous work). In this architecture, the PKG does not see Alice's identity (i.e., her e-mail address), which provides some measure of privacy compared with the original IBE architecture. Thus, the PKG computes the IBE private key K_A, but does not know who this key is for (and so cannot trivially decrypt ciphertexts intended for Alice). However, although it is an improvement over the original IBE, this scheme suffers from two potential weaknesses. First, the PKG learns the point P and so it can potentially try hashing the e-mail addresses of all known users in the environment until it finds a match for P, thus breaking Alice's anonymity. Second, if the ICA is not *honest* (in particular, if the ICA is *malicious*), then the PKG can collude with the ICA to discover which

user created a credential in which the attribute value a_1 is equal to $\hat{e}(P, P)$, thus breaking Alice's anonymity.

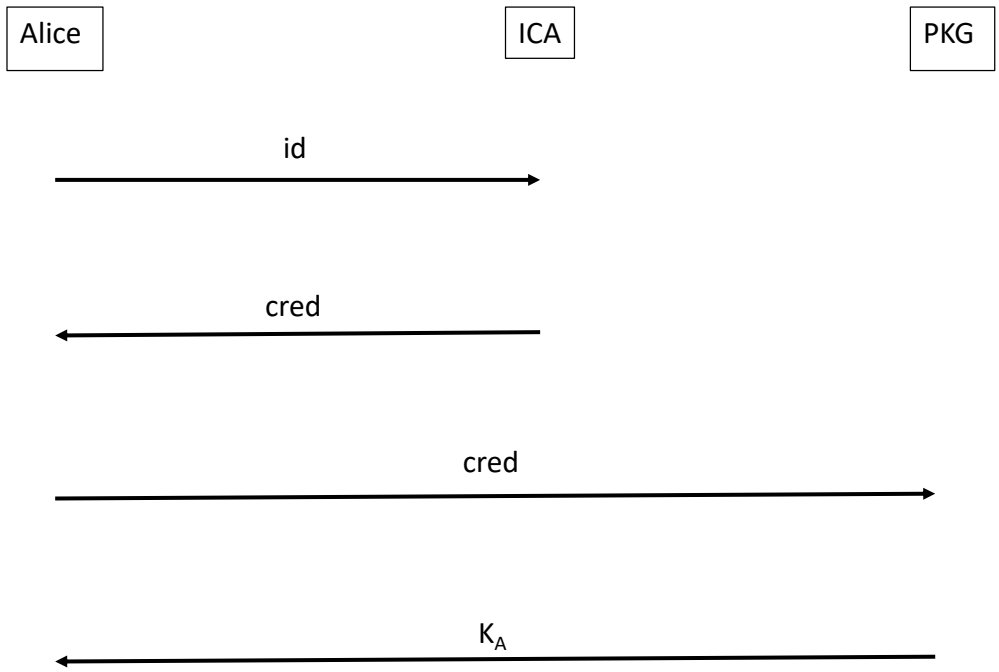

Figure 1. Basic scheme. Alice obtains her credential from the ICA, and subsequently uses this credential to obtain her private key from the PKG. Alice's privacy is lost if the PKG and ICA collude.

The augmented scheme given next mitigates these two weaknesses.

4.2. Augmented Scheme

Building on the framework above, we present an augmentation that protects against the weaknesses described. Instead of a single ICA, assume that there are v ICAs (ICA_1, ..., ICA_v). Furthermore, we define two ICA services—an *IBE Credentialing Service* and an *IBE Pseudonymizing Service*. For full generality, we specify that each ICA is able to provide either service upon request. The *IBE Credentialing Service* is identical to what was described previously: Alice proves ownership of her e-mail address and the ICA (blindly) signs a credential in which the first attribute a_1 is equal to ξ, where $\xi = \hat{e}(P, P)$ and $P = H_1$("*alice@gmail.com*").

For the *IBE Pseudonymizing Service*, Alice sends her credential public key h, the signature on that public key, a point Q, and a random integer $s \in \mathbb{Z}_q$. Using the *showing protocol*, Alice reveals attributes a_1 and a_2 in h. This ICA verifies the signature, confirms that a_2 is the identifier for the ICA whose public key verifies the signature on h, and confirms that a_1 is equal to $\hat{e}(Q, Q)$. This ICA then issues a new credential whose attribute a_1 is $\hat{e}(sQ, sQ)$. Note that we can think of this as an *IBE Anonymizing Service* or an *IBE Randomizing Service* if Alice will only use the resulting credential once, but *IBE Pseudonymizing Service* is a more suitable name if Alice might use the credential multiple times (for example, to obtain her private key from the PKG whenever she needs it rather than incurring the risk of storing the private key in her local environment).

The process is therefore as follows. Alice chooses an ICA at random, say ICA_i, and requests its *Credentialing* service, proving ownership of her e-mail address and obtaining a credential in which a_1 is $\hat{e}(P, P)$. She then chooses another ICA at random, say ICA_j, and requests its *Pseudonymizing* service, submitting P and a random value s_1, and obtaining a credential in which a_1 is $\hat{e}(s_1P, s_1P)$. She then chooses another ICA at random, say ICA_k, and requests its *Pseudonymizing* service, submitting s_1P and a random value s_2, and obtaining a credential in which a_1 is $\hat{e}(s_2s_1P, s_2s_1P)$.

Alice can now connect with the PKG. Alice presents her final credential with its signature and the point $R = s_2 s_1 P$, and she reveals attributes a_1 and a_2. The PKG verifies the signature, confirms a_2, and confirms that $\hat{e}(R, R)$ is equal to a_1 (The PKG is therefore convinced that this "random" point R is validly associated with the user presenting this credential). The PKG computes the private key $K = tR$ and sends K to Alice. Since $K = t(s_2 s_1)P$, Alice is able to compute her true private key $K_A = wK$, where $w = (s_2 s_1)^{-1}$ *mod* q, and decrypt any ciphertext created by encrypting a message using her public key P.

(It should be clear that each of Alice's connections with the ICAs for the *Pseudonymizing* services, as well as her connection with the PKG to obtain the key K, must be done over an anonymous channel (such as Tor [7]) so that her identity is not revealed to the party with whom she interacts or to any other observers. Note, however, that this does not remove the need for the Brands' issuing protocol to create a blind signature for the public key so that the ICA cannot recognize the resulting credential when Alice subsequently uses it; this is because the final segment of a Tor connection typically sends data in plaintext form).

With this augmented scheme, the PKG learns the elliptic curve point R, which is a randomized (i.e., blinded) multiple of P. Thus, the PKG will gain nothing by hashing e-mail addresses of known users since (with overwhelming probability) none of these will match the point R (in fact, if any e-mail address *does* happen to match, it will not be Alice's, since her e-mail address hashes to P, not to $s_2 s_1 P$). Furthermore, the PKG will learn that the credential presented by Alice was issued by ICA_k, but colluding with ICA_k will not reveal Alice's identity since ICA_k only saw the randomized point $s_1 P$ which (with overwhelming probability) does not match the hash of the e-mail address of any known user (and, as above, definitely will not match the e-mail address of Alice).

Given that the credential presented to ICA_k was issued by ICA_j, and that ICA_j saw Alice's identity point P, the PKG could break Alice's anonymity by colluding with *both* ICA_k and ICA_j. Clearly, however, colluding with two ICAs has a lower chance of success than colluding with a single ICA (since there is a lower probability that they are both corrupt). Furthermore, Alice can choose to use the *Pseudonymizing* service of any number η of ICAs (she can also visit the same ICA more than once at different times in her chain of pseudonymizations): the use of a digital credential as the blind token issued by an ICA, and the use of the bilinear map of an elliptic curve point as an attribute in the credential, allows Alice to easily obtain an unlimited number of pseudonymizations. Alice's anonymity will be retained if *any* of the ICAs in the chain is not corrupt (i.e., is *honest* or *honest-but-curious*); see Figure 2. Alice can thus reduce the probability that the PKG will learn her identity to an arbitrarily small value—she simply needs to keep track of all the random s_i that she chooses in order to properly unblind the private key K that she obtains from the PKG.

With digital credentials, it is also possible to require that a public key h must be signed by more than one ICA in order to form a valid credential or pseudonym. The various signers can chosen by a specified pseudorandom process based on h (for example, each signature determines who the subsequent signer must be). Requiring multiple signatures prevents a corrupt ICA from creating a credential or pseudonym by itself and using it to impersonate Alice in an interaction with the PKG in order to learn Alice's private key. As with the chain of pseudonymizations, the probability that a corrupt ICA will successfully impersonate Alice can be reduced to an arbitrarily small value by appropriately setting the required number of signatures on h. Therefore, a system administrator can choose an appropriate number of signatures per credential, δ, to protect against any estimated risk of corrupt ICAs in the environment.

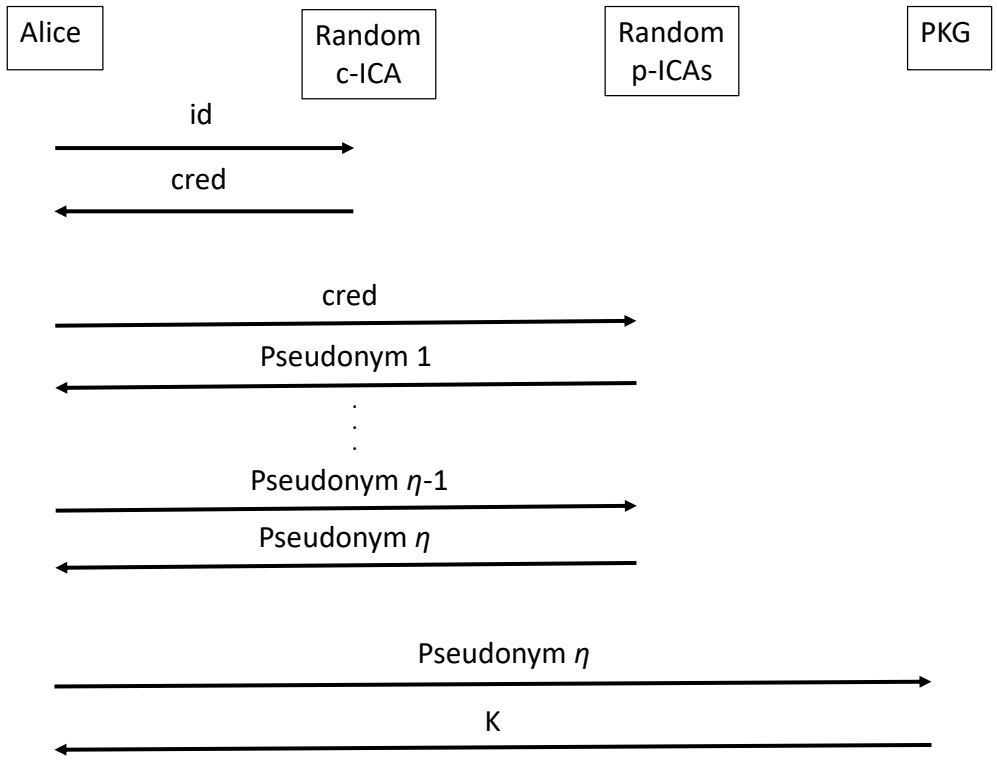

Figure 2. Augmented scheme. Alice obtains her credential from a credentialing ICA, and pseudonymizes it η times. She uses the final pseudonym to obtain her blinded key from the PKG; she can then unblind this to obtain her private key. Alice's privacy is preserved if any of the η pseudonymizing ICAs is not corrupt.

5. Discussion

The augmented scheme above mitigates the privacy concerns of the basic scheme (and of previous proposals). Even if the PKG knows the identity (i.e., the e-mail address) of every user in the environment, it cannot do a search to learn which user is associated with the private key K it has computed (because Alice's e-mail address will not hash to the randomized elliptic curve point that Alice submitted). Furthermore, it can only break Alice's anonymity if *every* ICA in Alice's chosen chain of pseudonymizations is corrupt (a probability that gets progressively smaller as the chain grows longer).

The original IBE scheme by Boneh and Franklin [2] requires Alice to prove her identity (i.e., to prove that she owns her e-mail address, as all schemes do) but, beyond this, no computation is required of Alice: the PKG computes Alice's private key and sends it to her. However, Alice has no privacy whatever if the PKG is *malicious* (or even if the PKG is *honest-but-curious* (HbC)). We can use these properties as a baseline against which to compare subsequent schemes.

- The scheme by Bendlin, et al. [5], is a threshold scheme over lattices in which τ out of n PKGs are required to compute Alice's private key, and the protocol will remain correct and secure even if up to $\tau' < \tau$ of the PKGs deviate from the protocol adversarially (for fixed system parameters τ and τ'). Alice must interact with τ PKGs to obtain her private key shares, but the (very minor) computation she requires is simply to combine the τ shares into her private key. Alice's privacy is lost if any τ PKGs are malicious (because they can collude to learn her private key).
- The scheme by Chow [4] is a separation scheme involving a PKG and an ICA. Alice receives a (blinded) certificate from the ICA and submits this to the PKG to acquire a blinded key. She then needs to unblind this value to obtain her private key.

- The scheme by Emura, et al. [6], is also a separation scheme involving a PKG and an ICA. Alice receives a (blinded) certificate from the ICA and submits this to the PKG to acquire a blinded key. She then needs to unblind this value to obtain her private key.
- Our basic scheme (Section 3 above) requires Alice to obtain a credential from the ICA, but then she can submit this credential and an elliptic curve point to the PKG without further computation.
- Our augmented scheme (Section 3 above) requires Alice to obtain a number of pseudonyms prior to contacting the PKG: there will be η pseudonyms and each pseudonym will have δ ICA signatures (the system administrator will choose δ for the environment, but Alice can choose the value η to reflect her desired level of privacy). Alice needs to be involved in all this computation, as well as in the unblinding of her private key from the PKG.

See Table 1 for a summary comparison. Note that we are focusing here on the additional computation that Alice must do in order to obtain her private key (i.e., computation beyond what is required in the original Boneh and Franklin IBE scheme). We are not considering any additional computation required by the authority nodes (the ICA(s) and the PKG) because these nodes are typically much more powerful than an individual user and thus do not generally impose constraints on usability from Alice's point of view. In [5], combining shares involves simply concatenating a collection of independent Gaussian samples and so requires very little computational effort from Alice. In [4], the unblinding step is a single exponentiation operation in a finite field. In [6], the unblinding step is the computation of the multiplicative inverse of an integer, a single exponentiation operation, and the multiplication of two integers, all in a finite field. In our proposals, credential signing and pseudonymization use precisely the issuing protocol shown in Figure 7 on page 18 of [9], and unblinding requires a single multiplication of an integer with an elliptic curve point (analogous to a single exponentiation in a finite field).

Table 1. Comparison of proposals.

Proposal	Setting	Computation by Alice to Obtain K_A	Privacy Breach
Boneh [2]	single PKG	none	PKG is HbC or malicious
Bendlin [5]	many PKGs (threshold scheme)	combining τ private key shares	τ PKGs are malicious (for fixed τ)
Chow [4]	single PKG, single ICA (separation scheme)	unblinding step	ICA & PKG are malicious
Emura [6]	single PKG, single ICA (separation scheme)	unblinding step	ICA & PKG are malicious
Basic scheme (this paper)	single PKG, single ICA (separation scheme)	credential signing	ICA & PKG are malicious
Augmented scheme (this paper)	single PKG, many ICAs (separation scheme)	credential signing, η pseudonymizations, δ signatures per pseudonym, unblinding step	$(\eta \times \delta)$ ICAs and PKG are all malicious (for fixed δ, user-chosen η)

It is clear from Table 1 that our augmented scheme requires the most computation on the part of the user, Alice. On the other hand, this scheme also provides the highest level of privacy since a large number of ICAs (as large as Alice wishes), along with the PKG, must *all* be malicious in order for the PKG to learn Alice's private key. The scheme by Bendlin, et al. [5], requires τ PKGs to all be malicious, but τ is a fixed system parameter that cannot be adjusted by Alice if she wishes to further reduce her risk of a privacy breach. The remaining schemes compromise privacy if at most 2 entities are malicious.

Note that in our augmented scheme Alice can request pseudonymizing services from her selected ICAs at any time: this can be done (long) before she connects with the PKG and, in fact, can be done when she initially joins the environment, (long) before any data has been encrypted for her.

5.1. Fully-Automated Services

It is worth noting that the ICAs can be implemented as fully-automated public-facing websites, operated as services independent from the PKG (that is, run by companies or private parties that are separate from the company that runs the PKG). This reduces the risk of collusion even further. In such a case, the system parameters ($p, q, E_\rho(a, b)$, and so on) would need to be known and used by all participants, but these parameters can be standardized values accepted by all parties.

A public-facing ICA can be fully automated because for the credentialing service it only needs to confirm the requester's e-mail address (which websites already do in countless automated registration scenarios today) and execute the credential issuing protocol. For the pseudonymizing service, it only needs to execute the credential showing and issuing protocols.

5.2. Smart Contract Instantiation

One interesting possibility is to separate the credentialing ICAs from the pseudonymizing ICAs, and to implement the pseudonymizing services as smart contracts [22,23] on a public blockchain (Note that the issuing protocol requires the ICA's private key for the credential signing step, but private values cannot exist in smart contract code and remain private. However, the signature step can be accomplished by having each smart contract do a mutually authenticated TLS connection to its own "credential signing website", sending the value which will be used as attribute a_1, and receiving the value r_0 which will be blinded by the user to r_0' later in the protocol. The signing website thus interacts with its authenticated ICA smart contract, but has no knowledge of the end user that initiated the issuing protocol). A smart contract implementation of each pseudonymizing ICA would have the benefit of ensuring confidence in the pseudonymizing service (because the code is open for inspection by anyone) and completely eliminating the possibility of collusion with the PKG (because the ICA code is open, fully automated, and immutable).

5.3. Potential Ubiquity

Public-facing ICAs can be available to anyone with an Internet connection (perhaps for free), whereas the PKG could conceivably be run as a commercial enterprise, charging a fee for the computation of an IBE private key. Alternatively, the PKG could be a private service available only to participants in a closed environment. On the other hand, if the PKG was also a (free) public-facing service, this would make IBE accessible to everyone, enabling the possibility of wide-spread, easy-to-use, secure transactions (e.g., secure e-mail and secure data transfer) between Alice and anyone who knows her e-mail address (Recall that this was the original vision of IBE but the "escrow problem" appears to have impeded its adoption).

5.4. Formalism and Security Analysis

The construction described above lends itself to a formalism, with security definitions and proofs, that is similar to the one given in Emura et al. [6]. The goal of the present paper

is to propose the architecture, explaining its various components, their functions, and their interactions. The formalism involves some detail and may be of interest primarily to a particular specialized audience. Thus, the extensive security analysis of this construction, which includes all the various entity types in the environment, models both variants (i.e., *honest-but-curious* and *malicious*) of each possible entity type, and exhaustively considers collusion among all possible collections of entity types, is presented in a separate companion paper [24].

The detailed security analysis given in [24] shows that choosing appropriate elliptic curve parameters, pseudonymizing chain length, and number of signatures for a valid credential/pseudonym, along with using strong techniques that are outside the scope of IBE algorithms and protocols (such as a trustworthy mechanism to prove ownership of an e-mail address to a credentialing ICA), can preclude almost all attacks against this construction. There is an attack that cannot be prevented: a malicious PKG learns Alice's identity by some means—such as by collusion with a malicious credentialing ICA or with a malicious user Bob who wants to breach Alice's privacy—and computes Alice's private key in order to read ciphertext that was intended for her. However, even this attack can be mitigated by taking explicit measures to ensure that the PKG is *not* malicious (for example, by implementing the PKG as a publicly-visible smart contract).

5.5. Limitations: Additional Computation and Revocation Needs

It is important to recognize that forming a chain of randomly-selected ICAs and keeping track of the random values s_i is additional work for Alice that will affect the usability of the system. Even if a client-side software tool does all this work for her, she will surely notice that the complete process of obtaining her IBE private key does not occur instantaneously (due to the additional computation required), which is a usability issue. Note that chain length, as well as number of signatures per credential, can be traded off against the risk of corrupt entities in the environment, so that the system implementer can choose a reasonable balance between extra computation and user privacy. In any case, if Alice chooses a pseudonymizing chain of length η and if each blind token requires δ digital signatures then, compared with a single ICA architecture, this construction will need an additional $\delta - 1$ signatures on the original credential and $\eta \times \delta$ total signatures to form the chain of pseudonyms. For numbers that would be reasonable for many environments (a chain length of 3–4, with 2–3 signatures per token), this corresponds to approximately an order of magnitude more computation for the signatures than the original IBE scheme. However, many users will only do this once at system set-up time (in order to obtain their private key) and never again unless they need to change their key pair for some reason, so this is not an on-going or per-transaction cost in any sense.

We also note that, although the trust traditionally required in the IBE PKG has been a long-standing concern, it is not the only concern associated with IBE systems. In particular, many researchers have proposed techniques to deal with revocation in IBE systems (for example, to avoid Alice having to switch to a new e-mail address if her IBE private key is compromised by an attacker); see [25–27]. Such techniques are entirely complementary to the scheme proposed in this paper, but would need to be considered in any real-world IBE deployment.

6. Conclusions

This paper proposes a scheme to reduce the trust required in the PKG for an identity-based encryption deployment. Using digital credentials and bilinear maps, our *basic* construction separates the PKG from an intermediate CA (ICA), reproducing both the functionality and the privacy level of the schemes in [4,6]. Our *augmented* construction, on the other hand, separates the PKG from a collection of ICAs and allows a user, Alice, to reduce the probability of any entity breaking her anonymity and learning her IBE private key to an arbitrarily small value.

The scientific contribution of our proposal is that it improves on previous constructions by providing greater deployability and a much higher degree of privacy to the user. The cost for this is increased computation for Alice and increased interaction with system authorities (i.e., ICAs), but this is a one-time cost when she needs to obtain her private key; it is not an on-going or per-transaction cost.

We hope that addressing the critical privacy issue with the PKG will lead to more widespread acceptance and deployment of IBE. However, we do recognize that other issues, such as a suitable revocation mechanism and standardization of system parameters, need to be resolved in conjunction with the reduction of trust in the PKG before such real-world deployments can occur.

Future work in this area will involve searching for ways to reduce Alice's overall computation when obtaining her private key (perhaps through developing computationally lighter blind signature protocols, designing efficient mechanisms to obtain multiple credential signatures in parallel, or determining optimal values of η and δ for given risk levels). We also encourage researchers to find the best ways to integrate the *augmented* scheme of this paper with efficient revocation techniques and with the system parameters of other security technologies with which it may need to interact.

Funding: This research was partially supported by Natural Sciences and Engineering Research Council of Canada (NSERC).

Data Availability Statement: Not applicable.

Conflicts of Interest: The author declares no conflict of interest. The funder had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Shamir, A. Identity-Based Cryptosystems and Signature Schemes. In *Advances in Cryptology—Proceedings of Crypto '84, LNCS*; Springer: Berlin/Heidelberg, Germany, 1985; Volume 196, pp. 47–53. [CrossRef]
2. Boneh, D.; Franklin, M. Identity-based encryption from the Weil pairing (extended abstract). In *Advances in Cryptology: Proceedings of Crypto 2001, LNCS*; Springer: Berlin/Heidelberg, Germany, 2001; Volume 2139, pp. 229–231. Available online: http://eprint.iacr.org/2001/090/ (accessed on 3 November 2022).
3. Boneh, D.; Franklin, M. Identity-Based Encryption from the Weil Pairing. *SIAM J. Comput.* **2003**, *32*, 586–615. [CrossRef]
4. Chow, S.S.M. Removing Escrow from Identity-Based Encryption: New Security Notions and Key Management Techniques. In *Public Key Cryptography—PKC 2009, LNCS*; Springer: Berlin/Heidelberg, Germany, 2009; Volume 5443, pp. 256–276. [CrossRef]
5. Bendlin, R.; Krehbiel, S.; Peikert, C. How to Share a Lattice Trapdoor: Threshold Protocols for Signatures and (H)IBE. In *Applied Cryptography and Network Security, LNCS*; Springer: Berlin/Heidelberg, Germany, 2013; Volume 7954, pp. 218–236. [CrossRef]
6. Emura, K.; Katsumata, S.; Watanabe, Y. Identity-based encryption with security against the KGC: A formal model and its instantiations. In *Theoretical Computer Science*; Elsevier: Amsterdam, The Netherlands, 2022; Volume 900, pp. 97–119. Available online: https://www.sciencedirect.com/science/article/pii/S030439752100699X (accessed on 3 November 2022).
7. The Tor Project. Available online: https://www.torproject.org/ (accessed on 3 November 2022).
8. Brands, S. *Rethinking Public Key Infrastructure and Digital Certificates: Building in Privacy*; The MIT Press: Cambridge, MA, USA, 2000.
9. Brands, S. A Technical Overview of Digital Credentials. *Credentica Paper*, 20 February 2002. Available online: http://www.credentica.com/overview.pdf (accessed on 3 November 2022).
10. Koblitz, N. Elliptic Curve Cryptosystems. *Math. Comput.* **1987**, *48*, 203–209. Available online: https://www.ams.org/journals/mcom/1987-48-177/S0025-5718-1987-0866109-5/S0025-5718-1987-0866109-5.pdf (accessed on 3 November 2022). [CrossRef]
11. Miller, V. Use of elliptic curves in cryptography. In *Advances in Cryptology—Proceedings of Crypto 1985, LNCS*; Springer: Berlin/Heidelberg, Germany, 1986; Volume 218, pp. 417–426. [CrossRef]
12. Galbraith, S.; Harrison, K.; Soldera, D. Implementing the Tate pairing. In *International Algorithmic Number Theory Symposium*; Springer: Berlin/Heidelberg, Germany, 2002; Volume 2369, pp. 324–337. [CrossRef]
13. Miller, V. The Weil pairing, and its efficient calculation. *J. Cryptol.* **2004**, *17*, 235–261. [CrossRef]
14. Barreto, P.S.L.M.; Costello, C.; Misoczki, R.; Naehrig, M.; Pereira, G.C.C.F.; Zanon, G. Subgroup Security in Pairing-Based Cryptography. In *Progress in Cryptology—LATINCRYPT 2015, LNCS 9230*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 245–265. [CrossRef]
15. Adams, C.; Lloyd, S. *Understanding PKI: Concepts, Standards, and Deployment Considerations*, 2nd ed.; Addison-Wesley: Boston, MA, USA, 2003.

16. Housley, R.; Polk, T. *Planning for PKI: Best Practices Guide for Deploying Public Key Infrastructure*; Wiley: New York, NY, USA, 2001.
17. Boneh, D.; Boyen, X. Secure Identity Based Encryption Without Random Oracles. In *Advances in Cryptology—Proceedings of Crypto 2004. LNCS*; Springer: Berlin/Heidelberg, Germany, 2004; Volume 3152, pp. 443–459. [CrossRef]
18. Gentry, C.; Peikert, C.; Vaikuntanathan, V. Trapdoors for Hard Lattices and New Cryptographic Constructions. Cryptology ePrint Archive, Paper 2007/432 (2007). Available online: https://eprint.iacr.org/2007/432 (accessed on 3 November 2022).
19. Gemmell, P. An introduction to threshold cryptography. In *CryptoBytes, a Technical Newsletter of RSA Laboratories*; RSA Laboratories: Redwood City, CA, USA, 1997; Volume 2, pp. 7–12.
20. Evans, D.; Kolesnikov, V.; Rosulek, M. *A Pragmatic Introduction to Secure Multi-Party Computation*; NOW Publishers: Delft, The Netherlands, 2018. Available online: https://www.cs.virginia.edu/~evans/pragmaticmpc/pragmaticmpc.pdf (accessed on 3 November 2022).
21. Gennaro, R.; Jarecki, S.; Krawczyk, H.; Rabin, T. Secure Distributed Key Generation for Discrete-Log Based Cryptosystems. *J. Cryptol.* **2007**, *20*, 51–83. [CrossRef]
22. Szabo, N. Smart Contracts: Building Blocks for Digital Markets. 1996. Available online: https://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart_contracts_2.html (accessed on 3 November 2022).
23. Levi, S.D.; Lipton, A.B. An Introduction to Smart Contracts and Their Potential and Inherent Limitations, Harvard Law School Forum on Corporate Governance. 2018. Available online: https://corpgov.law.harvard.edu/2018/05/26/an-introduction-to-smart-contracts-and-their-potential-and-inherent-limitations/ (accessed on 3 November 2022).
24. Adams, C. Security Analysis of a Privacy-Preserving Identity-Based Encryption Architecture. *J. Inf. Secur. (Spec. Issue Cryptogr. Netw. Secur.)* **2022**, *13*, 323–336. [CrossRef]
25. Boldyreva, A.; Goyal, V.; Kumar, V. Identity-based encryption with efficient revocation. In Proceedings of the 15th ACM Conference on Computer and Communications Security—CCS '08, Alexandria, VA, USA, 27–31 October 2008; p. 417. [CrossRef]
26. Elashry, I.; Mu, Y.; Susilo, W. Generic Mediated Encryption. In *Security and Privacy in Communication Networks*; Zia, T., Zomaya, A., Varadharajan, V., Mao, M., Eds.; Springer International Publishing: Berlin/Heidelberg, Germany, 2013; Volume 127, pp. 154–168. [CrossRef]
27. Seo, J.H.; Emura, K. Revocable hierarchical identity-based encryption. *Theor. Comput. Sci.* **2014**, *542*, 44–62. [CrossRef]