Scientific Research Publishing

Journal of Information Security, 2023, 14, 304-329
https://www.scirp.org/journal/jis
ISSN Online: 2153-1242
ISSN Print: 2153-1234

Construction and Implementation of a Privacy-Preserving Identity-Based Encryption Architecture

David Bissessar, Carlisle Adams

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

How to cite this paper: Bissessar, D. and Adams, C. (2023) Construction and Implementation of a Privacy-Preserving Identity-Based Encryption Architecture. *Journal of Information Security*, **14**, 304-329.
https://doi.org/10.4236/jis.2023.144018

Received: June 23, 2023
Accepted: August 28, 2023
Published: August 31, 2023

Abstract

A recent proposal by Adams integrates the digital credentials (DC) technology of Brands with the identity-based encryption (IBE) technology of Boneh and Franklin to create an IBE scheme that demonstrably enhances privacy for users. We refer to this scheme as a privacy-preserving identity-based encryption (PP-IBE) construction. In this paper, we discuss the concrete implementation considerations for PP-IBE and provide a detailed instantiation (based on q-torsion groups in supersingular elliptic curves) that may be useful both for proof-of-concept purposes and for pedagogical purposes.

Keywords

Identity-Based Encryption (IBE), Digital Credentials (DC), Privacy, Pairing-Based Cryptography, Supersingular Elliptic Curve, q-Torsion Group

1. Introduction

This paper describes, in some details, the considerations involved in implementing the Privacy-Preserving Identity-Based Encryption (PP-IBE) scheme presented by Adams [1] [2]. In 2001, Boneh and Franklin proposed the first efficient construction of Identity-Based Encryption (IBE) [3] [4], but their construction has a system authority, the Private Key Generator (PKG), that computes all user private keys; the PKG can therefore decrypt all ciphertexts in the environment. Subsequent constructions to reduce the trust in the PKG have relied on unrealistic assumptions (see [1]) and have not been described and implemented using easy-to-analyze numerical values.

The recently introduced proposal by Adams [1] [2] incorporates pseudonyms into the IBE process. This ensures that the PKG no longer needs to be fully trusted

(in particular, the PKG is successfully able to learn any given user's private key with probability as small as the user wishes). Adams' proposal is based on the IBE scheme designed by Boneh and Franklin (BF-IBE) [3] [4] but also uses the Digital Credentials (DC) technology of Brands [5] [6] for identity and pseudonym credentials.

The integrated protocol combines pairing-based cryptography (in the IBE portions) with discrete-log-based cryptography (in the DC portions); thus, parameter values must be carefully chosen to provide a consistent security level throughout the PP-IBE architecture.

This present paper provides the first concrete construction and numeric example (produced by our SageMath [7] software implementation[1]) of the complete Adams proposal. Our construction uses a q-torsion group (a cyclic subgroup of order q) of a supersingular elliptic curve over F_p, where p is a prime congruent to 2 modulo 3.

Section 1 of this paper presents an overview of PP-IBE. Section 2 provides a brief introductory background on BF-IBE, DC, PP-IBE, and the mathematical concepts supporting the proposed construction. Section 3 describes the construction itself. Section 4 presents a detailed numerical example using relatively small, easy-to-verify numbers. Section 5 provides some discussion, including suggested parameters for 128-bit security and enhancements for an admissible encoding of user identities. Section 6 presents performance measurements for the 128-bit secure version, and Section 7 concludes the paper.

1.1. Protocol Overview

Whereas the protocol of BF-IBE includes steps for entity setup, encryption, key extraction, and decryption, PP-IBE introduces a sequence of identity and pseudonym credential steps into the workflow. To obtain the keying material used for decryption from the PKG, Alice must redeem a valid identity/pseudonym credential. Identity and pseudonym credentials are obtained through interaction with a community of Intermediate Certification Authorities (ICAs) who each verify Alice's identity or the derivation of a pseudonym and issue a Brands digital credential. In the key extraction protocol, the PKG returns initial keying material which can be finalized ("un-blinded") only by Alice using random data produced during pseudonym creation. The overall flow, from encryption, through pseudonymization, to key extraction, and finally to decryption, is shown in **Figure 1**.

1.2. Protocol Steps

Setup: First, as a precursor to protocol execution, all participants undergo a SETUP phase. The required public primitives and parameters for PP-IBE, BF-IBE, and DC are initialized into the environment.

Following the environment setup, each of the scenario participants is set up. Thus, individual participants (Alice and Bob) and the service providers (ICA_i,

[1]PP-IBE SageMath implementation: [ada22] Integrated example 2 mod 3 v08.sage.

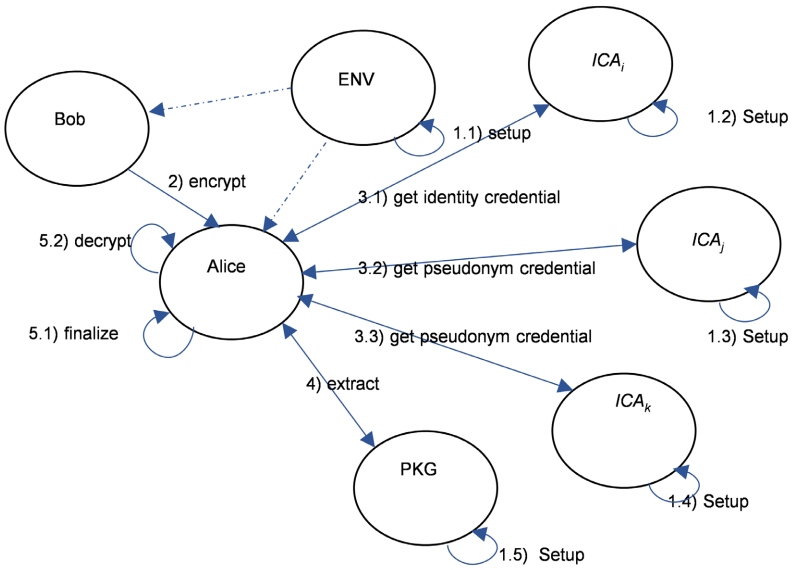

Figure 1. PP-IBE employs a five-step protocol. (1) System parameters are established for all participants. (2) Bob encrypts a message and sends it to Alice. (3) Alice engages with a community of Intermediate Certification Authorities (ICAs) which grant her identity and pseudonym credentials. (4) To obtain a decryption key, Alice presents a pseudonym credential to the PKG which validates it and provides her with initial keying material. (5) After finalizing her decryption key, Alice decrypts her message.

ICA_j, ICA_k and PKG) are initialized, granted access to the environment, and instantiated with public keys, private keys, and attributes, as appropriate.

Encrypt: Bob encrypts a message for Alice as in BF-IBE. Given an identity string for Alice (say, alice@gmail.com) and only public functions and data, Bob encrypts a message for Alice and sends her the resulting ciphertext.

Get Identity/Pseudonym Credentials: To decrypt the message received from Bob, Alice requires a decryption key. This key is calculated by Alice using keying material provided by the PKG and using a set of random values which she chooses and stores during pseudonym creation (each random value is shared with only a single ICA). Figure 1 shows an identity credential being issued by ICA_i followed by consecutive steps with ICA_j and ICA_k in which pseudonym credentials are elaborated. Thus, each pseudonym is associated with a random data value; Alice creates and retains these values in local protected storage.

Extract: Alice presents a pseudonym credential to query the private key generator (PKG) which conducts a verification protocol. On successful verification, the PKG calculates keying material specific to messages encrypted for Alice. This keying material cannot be used directly to decrypt the ciphertext sent by Bob. Rather, Alice must finalize ("unblind") the keying material into her decryption key by applying the random values accumulated during the pseudonym creation steps.

Decrypt: After finalization of her private key, Alice uses it to decrypt the ciphertext received from Bob. Decryption proceeds as specified in BF-IBE.

2. Background

This section introduces some of the background mathematical concepts used in PP-IBE, as well as the BF-IBE, DC, and PP-IBE algorithms themselves.

2.1. Fundamental Principles

Elliptic Curves. An elliptic curve over a finite field E/F_p can be seen as the set of (x, y) points with $x, y \in F_p$ which satisfy an equation of the form E: $y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$, where $x, y, a_1, a_2, a_3, a_4, a_6 \in F_p$.

The set of points in the curve $E(F_p)$ includes a distinguished point, the point at infinity $\mathcal{O}$, and forms a group under addition. The number of points on a curve $\#E(F_p)$ is called the order of the curve. The Hasse theorem places a bound on the number of points in a curve: $\#E(F_p) = p + 1 - t$, where $|t| \le 2\sqrt{p}$.

Background on supersingular elliptic curves can be found in [8]. Given a prime base b, an integer exponent $e \ge 1$, and t in a range, such that $|t| \le 2\sqrt{p}$, a supersingular curve $E(F_p)$ over a field F on prime power $p = b^e$ is a curve such that its order $\#E(F_p) = p + 1 - t$, with $b \mid t$ [9] [10]. The worked example of §4, uses the curve $E(F_{281}): y^2 = x^3 + 1$, which has order 282, with $p = b = 281$, $e = 1$ and $t = 0$. (To keep our example as simple as possible, we want to use a modulus that is a prime, rather than a prime power; thus, $e = 1$ and therefore $p = b$. Furthermore, supersingular curves of the form $y^2 = x^3 + 1$ over F_p are known to have order $p + 1$, meaning that $t = 0$. We chose $p = 281$ as this is a relatively small prime that is not too trivial such as 7 or 13.)

Each point on a curve also has an order, a scalar r, the number of times that point must be added to itself to give $\mathcal{O}$: the order of a point $o(P) = r$, so that $r * P = \mathcal{O}$ for $P \in E(F_p)$. The r-torsion subgroup of a curve $E(F_p)[r]$ is the subset of points in $E(F_p)$ which have order r.

In this paper, curves are used in the context of bilinear maps ("pairings"). A taxonomy of pairing-friendly curves (including FMT curves, GMV curves, Freeman curves, Cyclotomic families, Sporadic families, Scott-Barreto families, Supersingular curves, Cocks-Pinch curves, MNT curves, and DEM curves) is presented in [11].

The construction given in this paper uses a supersingular curve of the form $y^2 = x^3 + 1$ over $F(p)$, with prime $p \equiv 2 \pmod 3$. This follows the construction in BF-IBE and lends itself well to the requirements of PP-IBE where a point p_1 must be paired with itself. Other curves are also possible, including other supersingular curves and MNT curves.

Bilinear Maps. A bilinear map is a function $\hat{e}: G_0 \times G_1 \to G_T$ where G_0, G_1 and G_T have order q. We focus on so-called *symmetric* pairings, where $G_0 = G_1$. The mapping is bilinear if $\forall P_1, P_2 \in G_1$ and $\forall a, b \in Z_q^*$, $\hat{e}(P_1^a, P_2^b) = \hat{e}(P_1, P_2)^{ab}$, and it is non-degenerate if $\forall$ non-trivial points $P_1 \in G_1$, $\hat{e}(P_1, P_1) \ne 1$ (the multiplicative identity element in G_T). Two common pairing functions are the Weil pairing and the Tate pairing [8] [12] [13]. The worked example in this document was verified with both the Weil and Tate pairings but, for the sake of

brevity, only the computations demonstrating the Weil pairing are shown in this paper.

The embedding degree of elliptic curve $E(F_p)$ with respect to the order q of its q-torsion subgroup is the smallest positive integer k such that $p^k \equiv 1 \pmod{q}$ (for background, see [8]). The embedding degree affects security. Curves with a small embedding degree are susceptible to the MOV reduction proposed by Menezes, Vanstone and Okamoto [9]. The MOV reduction allows the discrete log problem to be translated from the elliptic curve setting of G_1 to the finite field setting of G_T, in which there are faster sub-exponential algorithms to solve it [9]. For this reason, sufficiently large security parameters must be selected when instantiating our construction of PP-IBE (see §3, §5).

Distortion Maps. As we shall see, in its derivation of ξ, PP-IBE requires that the two inputs to the pairing be from the same cyclic group G_1. The Weil pairing produces the degenerate result 1 when the inputs P and Q are linearly dependent. The Tate pairing also has this property when $k > 1$. One technique for working around this issue is to use a supersingular curve, E, with a distortion map φ. The distortion map projects points from the base curve onto the curve on the extension field, in a manner that the points are no longer linearly dependent, and so the result of the Weil pairing is non-degenerate [3] [9] [12]. When distortion map functionality is required, supersingular curves are not the only choice. MNT curves and their trace map may also be used [14]. We selected the supersingular curve for pedagogical reasons, as a natural progression from BF-IBE and PP-IBE.

2.2. Boneh-Franklin Identity-Based Encryption

In 2001, Boneh and Franklin published the first Identity-based Encryption scheme [3]. Their scheme is defined in terms of bilinear maps. BF-IBE features four protocols.

Setup: The environment is initialized with public parameters q prime, groups G_1 and G_T of order q, bilinear map $\hat{e}: G_1 \times G_1 \to G_T$, generator $G \in G_1$, and hash functions $H_1: \{0,1\}^* \to G_1^*$, where $G_1^* = G_1 \setminus \{\mathcal{O}\}$, $H_2: \{0,1\}^n \times \{0,1\}^n \to Z_q^*$, $H_3: G_T \to \{0,1\}^n$, and $H_4: \{0,1\}^n \to \{0,1\}^n$. The PKG is given master private key $t \in Z_q^*$ and master public key $T = tG$.

Encrypt: Message $M \in \{0,1\}^n$ is encrypted to ciphertext (u, v, w) using Alice's public identity string $ID_i \in \{0,1\}^n$ as follows.

1) Map the identity string to point $I_A = H_1(ID_i)$.
2) Apply the pairing $\mu = \hat{e}(I_A, T)$.
3) Select random $\sigma \in \{0,1\}^n$.
4) Set exponent r to $H_2(\sigma \| M)$.
5) Compute $z = \mu^r$.
6) Compute ciphertext $(u, v, w) = (rG, \sigma \oplus H_3(z), M \oplus H_4(\sigma))$.

Extract: To decrypt a message the recipient must present their ID_i to a Private Key Generator (PKG). The algorithm first maps the string to a group point $I_A = H_1(ID_i)$ and then multiplies the point by the master private key to pro-

duce the decryption key $K_A = tI_A$. The decryption key K_A is returned to the caller.

Decrypt: The decryption algorithm applies the private key K_A to ciphertext (u, v, w) to obtain plaintext M.

1) Compute $z = \hat{e}(K_A, u)$.

2) Compute $\sigma = v \oplus H_3(z)$.

3) $M = w \oplus H_4(\sigma)$.

4) Verify that $u == rG$: if these are equal, return M; if they are not equal, the ciphertext is rejected.

Boneh and Franklin define an adaptive chosen ciphertext notion of security IND-ID-CCA applicable to identity-based encryption, and three different variations of their protocol.

2.3. Digital Credentials

Brands Digital Credentials [5] [6] may be described in terms of three protocols: "setup", "issue" and "show".

Setup: During setup, the environment is initialized with p, q, g_0, where p is a large prime, q is the prime order of a multiplicative subgroup of Z_p^*, and g_0 is a generator of this q-order subgroup. Issuers of Digital Credentials are each initialized with a private key $(y_1, y_2, \cdots, y_m, x_0)$ where $y_1, y_2, \cdots, y_m, x_0 \in Z_q$ and a public key $(g_1, g_2, \cdots, g_m, h_0)$, where $g_i = g_0^{y_i}$ for i in [1, m] and $h_0 = g_0^{x_0}$.

Issue: The issue protocol proceeds between an individual, Alice, and a credential issuer, say Bob, as follows.

1) Alice selects $\alpha \underset{R}{\leftarrow} Z_q$ and calculates $h = \left(g_1^{x_1} g_2^{x_2} \cdots g_m^{x_m} h_0\right)^{\alpha} \bmod p$ using Bob's public key and her attributes $(x_1, x_2, \cdots, x_m)$. Alice retains h as the first part of the credential.

2) Bob selects $w_0 \underset{R}{\leftarrow} Z_q$ and calculates $a_0 = g_0^{w_0} \bmod p$ which he sends to Alice.

3) Alice selects $\beta, \gamma \underset{R}{\leftarrow} Z_q$ and calculates the second credential component $c_0' = H(h \mid h_2)$ where $h_2 = g_0^{\beta}\left(\left(g_1^{x_1} g_2^{x_2} \cdots g_m^{x_m} h_0\right)^{\gamma}\right) a_0 \bmod p$. She sends a blinded version $c_0 = (c_0' - \beta) \bmod q$ to Bob for his signature.

4) Bob creates receives c_0 and creates signature material $r_0 = (w_0 - c_0) / (x_0 + x_1 * y_1 + x_2 * y_2 + \cdots + x_m * y_m) \bmod q$ and sends this r_0 to Alice.

5) Alice evaluates an issuance verification relation, confirming that $g_0^{c_0}\left(g_1^{x_1} g_2^{x_2} \cdots g_m^{x_m} h_0\right)^{r_0} \bmod p$ is equal to a_0. If valid, she finalizes Bob's signature to obtain $r_0' = (r_0 + \gamma) / \alpha \bmod q$ and stores the issued credential (h, c_0', r_0') for later use in the showing protocol.

Show: Alice shows selected contents of her credential to verifier Victor as follows.

1) Alice sends the credential (h, c_0', r_0') to Victor.

2) Victor confirms the credential is valid by evaluating a verification relation, confirming that c_0' is equal to $H\left(h \mid g_0^{c_0'} h^{r_0'} \bmod p\right)$.

3) If the verification relation passes, Alice and Victor engage in a proof of knowledge protocol (a verification of a Boolean predicate on Alice's credential attributes). Brands describes a variety of predicates in which Alice reveals the value of a required attribute and proves knowledge of the remaining attributes in the credential [5] [6]. One such predicate will be presented in the construction section of this document.

4) Following successful proof of knowledge, Victor provides Alice with a service or access to a resource. In the PP-IBE protocol, after integrity verification and proof of knowledge, the ICA in the role of verifier grants a pseudonym credential.

Brands [6] presents variations of the protocols including a version based on the RSA problem. This paper restricts itself to the protocol as described on the discrete logarithm problem [5] [6]. Brands digital credentials are used by Adams to sign and certify identity and pseudonym credentials. The nomenclature in our construction uses p' and q' for p and q above for integration with BF-IBE (which itself uses a p and q).

2.4. Privacy-Preserving Identity-Based Encryption

In [1] [2], Adams proposes PP-IBE, an approach to reduce the amount of trust that must be placed in the PKG. PP-IBE uses digital credentials as both identity certificates and pseudonyms, and proposes a community of ICAs who certify these credentials. In this paper, we focus on the "augmented scheme" of PP-IBE in which Alice interacts with a community of ICAs to obtain a final pseudonym credential that is exchanged for keying material with the PKG.

Setup: The setup protocol of PP-IBE combines the setup activities of BF-IBE and DC, without introducing additional global (*i.e.*, publicly accessible) data. Thus PP-IBE requires, for BF-IBE, a curve $E_\rho(a,b)$ with generator point G which generates the group G_1, and a bilinear map $\hat{e}: G_1 \times G_1 \to G_T$. For DC, PP-IBE requires primes p' and q', and g_0, a generator of the q'-order subgroup of $Z_{p'}^*$.

Encrypt: Encryption proceeds as defined by BF IBE, thus using $I_A = H_1$ (id_string) and $\mu = \hat{e}(I_A, T)$, producing ciphertext (u,v,w) which is sent to Alice. Alice requires the help of a community of ICAs and a PKG to decrypt the ciphertext.

Identity Credential Issuance. Alice obtains an identity credential from ICA_i as follows. 1) Alice sends her id_string to ICA_i who verifies Alice's ownership. 2) ICA_i computes $p_1 = I_A = H_1$ (id_string) and $\xi_{p_1} = \hat{e}(p_1, p_1)$. Alice and ICA_i engage in the Brands Issuing protocol with $x_1 = \xi_{p_1}$ and x_2 = the id of ICA_i. On successful completion of the issuing protocol, Alice holds $cred_i = (h, c_0', r_0')$, a signed identity credential enclosing a ξ_{p_1} certified by a trusted service provider.

Pseudonym Credential Issuance. Alice may choose to pseudonymize the identity credential issued by ICA_i. To do this, Alice selects another Intermediate

Certification Authority, say ICA_j and proceeds as follows. 1) Alice sends $cred_i$, p_i, $attr_i = (a_1, a_2)_i$ and s_j to ICA_j. 2) ICA_j applies the verification relation to confirm digital integrity of the credential. 3) If the credential passes the integrity check, Alice and ICA_j conduct a Brands proof of knowledge on attributes $a_1 = \xi_{p_1}$ and $a_2 = id_i$. 4) ICA_j verifies provenance, confirming that that $\hat{e}(p_i, p_i) == a_{1_i}$ 5) Following successful verification of $cred_i$, ICA_j computes pseudonym point $p_j = p_i * s_j$, $\xi_j = \hat{e}(p_j, p_j)$ and proceeds with the Brands issuing protocol with $attr_j = (a_1, a_2)_j = (a_{1_j}, a_{2_j}) = (\xi_j, id(ICA_j))$.

Alice may now choose to pseudonymize further: she can present $cred_i$, p_i, s_i to another authority (say, ICA_k), proving knowledge of $attr_j$. On successful verification, pseudonym $p_k = s_j * p_j$ is created by ICA_k, and a new credential, $cred_k$, is issued to Alice. Using this process, Alice may create a chain of credentials on pseudonyms of any length she wishes. Note that the random values s_i, s_j, ⋯ are retained by Alice for use in key extraction.

Key Extraction. In PP-IBE, key extraction is in two parts. First Alice provides a credential on a pseudonymized point p_k to the PKG. After verifying the credential and Alice's proof of knowledge of the signed attributes, the PKG creates keying material $K = t * p_k$, where t is the PKG's master secret key. This initial keying material is sent to Alice who creates the decryption key K_A by multiplying this K with the product of the inverses of the scalars used to create p_k, thus $K_A = wK$, where $w = \prod_{s_i \in S} s_i^{-1} (\mathrm{mod}\, q)$.

Decryption. After having successfully created K_A, Alice can decrypt ciphertext (u, v, w) as per the BF-IBE algorithm.

Note that in [1] [2], PP-IBE uses two attributes within the credential (one for ξ and one for the id of the issuing ICA). However, in an actual implementation, the output of the pairing function (either the Weil or Tate pairing) will be a polynomial in the extension field p^k. For the supersingular curves on which we base our construction below, $k = 2$ and the pairing output will be a polynomial with two coefficients. Therefore, the credentials in our construction have been modified to contain three attributes (two for the coefficients of the pairing output and one for the id of the issuing ICA).

3. Construction

3.1. PP-IBE Group Parameters

The parameter selection consists of finding a tuple of primes (q, p, q', p'). These values determine the main group sizes used in our construction PP-IBE. Parameter q sets the size of G_1; in our construction G_1 is q-torsion group of the base elliptic curve. Parameter p creates F_p, the prime field upon which the base elliptic curve is defined. Parameter p' creates $Z_{p'}$ the base field for digital credentials. Parameter q' is the order of the multiplicative subgroup of $Z_{p'}^*$; digital credential attributes and other exponents are drawn from $Z_{q'}^*$.

The parameter selection algorithm is as follows:

1) Select q a prime.
2) Find p prime such that $q \mid p+1$, $q^2 \nmid p+1$ and $p \equiv 2 \bmod 3$.
3) Set q' to p.
4) Find p' prime such that $q' \mid p'-1$.
5) Return (q, p, q', p').

3.2. Definition of Mathematical Objects

Given parameters (q, p, q', p') initialize the required mathematical objects:

1) Let p define the prime field F_p.
2) Set base curve $E\left(F_p\right): y^2 = x^3 + 1$.
3) Set G_1 to be the torsion group $E(F_p)[q]$.
4) Extension field F_{p^2}, curve on extension field $E\left(F_{p^2}\right): y^2 = x^3 + 1$, and corresponding torsion group $E\left(F_{p^2}\right)[q]$ are created.
5) Set G_1' to be $E\left(F_{p^2}\right)[q]$.
6) Set $Z_{q'}^*$, $Z_{p'}$, $Z_{p'}^*[q']$.
7) Return $(E\left(F_p\right)[q], E\left(F_p\right), E\left(F_{p^2}\right), Z_{q'}^*, Z_{p'})$.

The base curve $E\left(F_p\right): y^2 = x^3 + 1$ where $p = 2 \bmod 3$ has the following properties. It is a supersingular curve of order $\#E(F_p) = p + 1$. Since $q \mid p + 1$, $E(F_p)$ contains a torsion group $E(F_p)[q]$ of order q. The curve $E(F_p)$ also supports a distortion map $\varphi(x, y) = (\zeta x, y)$ where $\zeta \in F_{p^2}$ and $\zeta^3 = 1$.

3.3. Construction of PP-IBE

The parameters (q, p, q', p') and mathematical objects $(E\left(F_p\right)[q], E\left(F_p\right), E\left(F_{p^2}\right), Z_{q'}^*, Z_{p'}^*[q'], Z_{p'})$ are used to initialize the components of PP-IBE as follows:

1) Set G_1 to $E\left(F_p\right)[q]$.
2) Set $f: E\left(F_p\right)[q] \times E\left(F_p\right)[q] \to E\left(F_{p^2}\right)$ to the Weil (or Tate) pairing.
3) Define $\hat{e}(p, q) \triangleq f\left(p, \varphi(q)\right)$ where φ is the distortion map of $E(F_p)$.
4) Set m, the number of attributes for digital credentials, to be 3.
5) Set $Z_{q'}^*$ to be the field from which digital credential attributes and exponents are drawn.
6) Set $Z_{p'}^*$ to be the base field of digital credentials.
7) Set G and G_{H_1} to be elements of G_1.
8) Set g_0 to be a generator of $Z_{p'}^*[q']$.

3.4. Hash Functions

Function $H_1: \{0,1\}^* \to G_1^*$ is implemented using G_{H_1}, a generator of G_1, which is multiplied by the SHA256 hash of the input string (this product is reduced modulo q). Function $H_2: \{0,1\}^n \times \{0,1\}^n \to Z_q^*$ takes the hash of the input strings, reduces it modulo $q - 1$ and adds one to the result. Function $H_3: G_T \to \{0,1\}^n$

has the polynomial representation of the arguments and extracts the n least significant bits. Function $H_4 : \{0,1\}^n \to \{0,1\}^n$ hashes the input string and extracts the n least significant bits.

3.5. Other Considerations

Use of Distortion Map. In PP-IBE, the identity token $\xi_{p_i} = \hat{e}(p_i, p_i)$ requires that point $p_i \in G_1$ be paired with itself. The Weil pairing $f(P,Q)$ returns the degenerate result 1 when the inputs are linearly dependent, which is precisely the case in determining ξ_{p_i}. The problem is addressed by using the distortion map to implement a modified pairing function $\hat{e}(p,q) = f(p, \varphi(q))$ where f is a pairing such as the Weil or Tate pairing. The distortion map is defined as $\varphi(x,y) = (\zeta x, y)$ where $\zeta \in F_{p^2}$ and $\zeta^3 = 1$. This map $\varphi : (F_p) \to E(F_{p^2})$ translates a point in $E(F_p)$ to a linearly independent point in $E(F_{p^2})$, which allows the pairing to be applied returning non-degenerate results.

Three-Base Credential. The modified pairing function uses the Weil or Tate pairing and returns $\xi \in F_{p^2}$, a degree-one polynomial with two coefficients F_p. Since $q' = p$, these coefficients may carried as attributes $Z_{q'}$ in the digital credential. There are different approaches to carrying ξ in the credential; however, we found that carrying both coefficients is convenient for calculations in key extraction. Thus, one attribute (and corresponding base) is added to PP-IBE as defined in [1]. Commensurate changes are required to the digital credential setup, issue, and verification protocols.

Choice of the Curve. A supersingular curve was chosen for two main reasons. First, the distortion map permits the pairing of p with itself. Second, this curve follows the construction given in [3]; our worked example thus complements [3] and contributes to its body of knowledge. Note that by choosing this curve, we also benefit from the BDH generator and security proofs given in [3].

Security Impact. From the perspective of elliptic curves and bilinear pairings, we refer to the discussion in [3]. The MOV reduction [9] can be applied to such supersingular curves as we have chosen for our construction. Thus as pointed out in [3], care must be taken to choose parameters such that the discrete log $G_1 = E(F_p)[q]$ remains difficult in $G_T = F_{p^2}$. In addition to security in elliptic curves and bilinear maps, we must also consider the difficulty of the traditional discrete log problem due to use of the digital credentials. Parameter sizes are discussed in §5.

Issue_on_Point(). We introduce a function issue_on_point() which takes a point in G_1 and a scalar and can be used for issuing both credentials and pseudonyms. Identity credentials and pseudonym credentials are both issued based on some ξ_i derived from a source point in G_1. The elegance of the algorithm allows one to be expressed in terms of the other. An "issue_on_point" algorithm has been implemented, which accepts a point and a scalar (as well as the other DC-required arguments) to produce a credential on the point resulting from the

multiplication of the point and the scalar received as arguments. This protocol is called both by identity issuance and pseudonym issuance logic. If an identity credential is to be issued, the scalar has the value one and multiplication leaves the point unchanged. If a pseudonym credential is to be issued, the scalar is the s_i value selected by Alice to create the pseudonym. See Figure 2 for an illustration of the steps involved in credential issuance.

4. Worked Example

The following demonstrates the augmented flow from [1] on sample parameters describing an elliptic curve of the form $E\left(F_p\right): y^2 = x^3 + 1$ where $p = 2 \bmod 3$. The numbers are based on output generated by the SageMath [7] implementation of PP-IBE[2].

4.1. System Parameter Generation

Group Parameters: First, group parameters are selected. We seek p, q, q', p' all prime with $q \mid p+1$ and $q^2 \nmid p+1$, $p = 2 \bmod 3$, $p \neq 3 \bmod 4$, $q' = p$, and $q' \mid p'-1$. Our worked example (a "toy" example with deliberately small numbers for readability and easy verifiability) uses $q = 47$, $p = 281$, $q' = 281$, and $p' = 563$.

Let $q = 47$, for a torsion group G_1 of size 47. Parameters o, p, q' and p' follow from this choice. Our construction uses a supersingular curve of the form $E\left(F_p\right): y^2 = x^3 + 1$, where $p = 2 \bmod 3$, which has order $o = \#E\left(F_p\right) = p+1$. We require p prime such that $q \mid p+1$ and $q^2 \nmid p+1$. For the worked example, we select $p = 281$, so that our base curve is E/F_{281}: $y^2 = x^3 + 1$ with $G_1 = E(F_{281})$ [47],

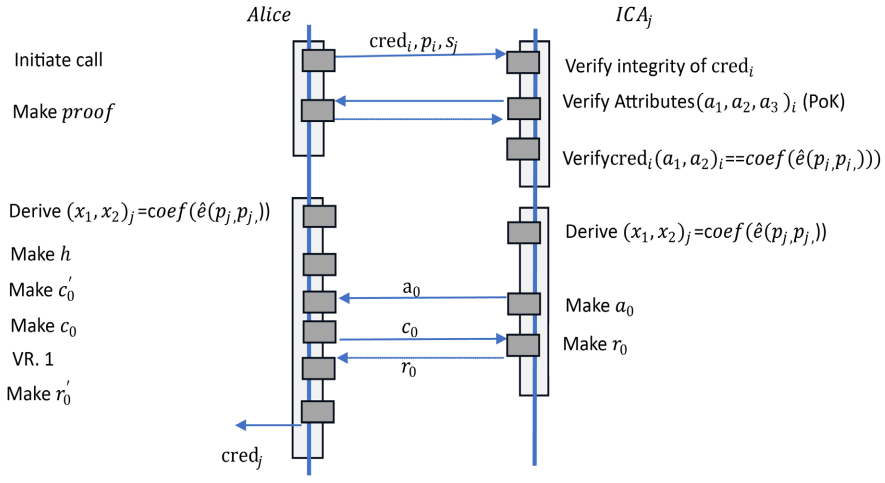

Figure 2. In the pseudonym service Alice submits credential $cred_i$, point p_i, and scalar s_j to ICA_j. First, $cred_i$ is verified for data integrity, attribute ownership and point correspondence. After verification, an issuance protocol is conducted to create pseudonym credential $cred_j$. The verification and Issuance protocols of Brands are augmented with point verification and derivation steps which use modified pairing $\hat{e}(.)$.

[2]Ibid.

the set of points in $E(F_{281})$ with order 47, along with $\mathcal{O}$. The order of the base curve $o = \#E(F_p) = 282$ is divisible by $q = 47$. The embedding degree of our worked example is $k = 2$, the smallest positive integer k such that $\#G_1 \mid \#E(F_p)^k + 1$. In our example, $47 \mid 282^k + 1$.

Credential Parameters: As per our construction, the output of the pairing function will be an element in the extension field F_{p^2}, a polynomial of degree one with coefficients in F_p. We carry the coefficients as attributes in the digital credential. The q' of the digital credential parameters must be large enough to accommodate these numbers. We set $q' = p = 281$. It remains only to find p', the modulus for digital credentials, such that $q' \mid p' - 1$. For our worked example, we select $p' = 563$.

Generators: Select generator $G = (1, 132)$ for IBE and $g_0 = 3$ for digital credentials.

4.2. Key Generation

The ICAs are each initialized with their DC key pairs in which private key $K_{priv} = (y_1, y_2, y_3, x_0)$ which are random values in $Z_{q'}$ and public key $K_{pub} = (g_1, g_2, g_3, h_0)$ where $g_i = g_0^{y_i} \bmod p'$ and $h_0 = g_0^{x_0} \bmod p'$. The PKG is initialized with a BF-IBE key pair in which the master secret key is a random scalar in Z_q, say $t = 23$, and the public key is calculated as $T = t * G = 23 * (1,132) = (252, 202)$. Select $\zeta = 68b + 106$ (here, $\zeta \in F_{p^2}$ and $\zeta^3 = 1$). Table 1 presents the key pairs for the service providers of the scenario presented in the augmented scheme of [1].

4.3. Encryption

Encryption proceeds according to BF-IBE. Here, ciphertext (u, v, w) is created for M = "*abc*" (msg_bits: "011000010110001001100011", of length $n = 24$) using Alice's identity string "alice@gmail.com".

First u is calculated. Alice's identity string is first mapped to point $I_A \in \mathbb{G}_1 = H_1$ ("alice@gmail.com") = (34, 33). Next the identity point I_A is paired with the master public key $\mu = \hat{e}(I_A, T)$. Recall that $\hat{e}(p_1, p_2) = f(p_1, \varphi(p_2))$ where $\varphi(x, y) = (\zeta x, y)$, $\zeta = 68b + 106$, $f()$ is either the Tate or Weil pairing, and b is

Table 1. Service provider key pairs.

Service Provider	Key Pair
ICA_i	K_{pub}: (243, 541, 498, 27) K_{priv}: (5, 9, 7, 3)
ICA_j	K_{pub}: (289, 326, 349, 470) K_{priv}: (15, 19, 17, 13)
ICA_k	K_{pub}: (68, 441, 49, 508) K_{priv}: (25, 29, 27, 23)
PKG	T: (252, 202) t: 23

an arbitrary symbolic variable to be used in polynomials in F_{p^2}. Using the Weil pairing, $\mu = \text{weil\_pairing}(I_A, \varphi(T)) = \text{weil\_pairing}((34,33), \varphi(252,202))$ $= \text{weil\_pairing}((34,33),(276b+17,202)) = 216b+147$. Let $\sigma =$ "100101000111100011000010", and $r = H_2(\sigma, \text{msg\_bits}) = 43$, then $u = r*G = 43*(1,132) = (263,167)$.

The second component of the ciphertext, v, is calculated as follows. Calculate $z = \mu^r = (216b+147)^{43} = 21b+60$ (where exponentiation is reduced modulo the Conway polynomial [15] [16] $b^2 + 280b + 3$) and let $h3z = H_3(z, n) =$ 001100110011011000110001; then component $v =$ sigma xor h3z = 101001110100111011110011.

The last component of the ciphertext is $w =$ msg_bits $\oplus H_4(\sigma)$. Assuming $H_4(\sigma)$ = 011001100011010001100010, then $w =$ 011000010110001001100011 $\oplus$ 011001100011010001100010 = 000001110101011000000001.

The ciphertext of $M =$ "*abc*" encrypted using Alice's identity string "alice@gmail.com" is (u,v,w) = ((263, 167), 101001110100111011110011, 000001110101011000000001).

4.4. Identity Credentials

PP-IBE uses interactions between Alice and ICAs to certify her identity and pseudonym as precursors to interacting with the PKG to obtain the decryption keys.

Alice engages with ICA_i to obtain a digital credential (h, c_0', r_0') certifying her identity. For this worked example, we choose the following values at random: (α = 12, $\beta = 6$, $\gamma = 2$, $w_0 = 8$). As a first step, Alice calculates h on her own. She calculates her identity point $p_1 = H_1$('alice@gmail.com') = I_A = (34, 33). Following this, she calculates $\xi_i = \hat{e}(p_i, p_i) = \text{weil\_pairing}(p_i, \varphi(p_i)) = \text{weil\_pairing}((34,33),$ $(64b+232,33)) = 13b+211$. Assuming service provider id = 27, Alice has attributes $(x_1, x_2, x_3) = (211,13,27)$. Assuming $\alpha = 12$, she calculates and retains $h = (g_1^{x_1} g_2^{x_2} g_3^{x_3} h_0)^{\alpha} \bmod p_1 = ((243^{211})(541^{13})(498^{27})(27))^{12} \bmod 563 = 256$.

As her next step Alice calculates $c_0' = H(h \mid h_2)$ with input a committed random from the ICA. ICA_i selects random w_0 (assume $w_0 = 8$), calculates $a_0 = g_0^{w_0} = 3^8 \bmod 563$, and sends a_0 to Alice. Alice uses a_0 to calculate $h_2 = g_0^{\beta}(g_1^{x_1} g_2^{x_2} g_3^{x_3} h_0)^{\gamma} a_0 \bmod p'$ = $3^6(243^{211} * 541^{13} * 498^{27} * 27)^2 * 368 \bmod 563$ = 99. This h_2 is combined with h to form $c_0' = H(h \mid h_2)$ = $SHA256(256|99)$ mod 281 = 204. This c_0' will be the second component in the digital credential.

To calculate r_0', Alice initiates by sending $c_0 = c_0' - \beta \bmod q'$ = 204 – 6 mod 281 = 198 to ICA_i. The ICA calculates $r_0 = (w_0 - c_0)/(x_0 + x_1 * y_1 + x_2 * y_2 + x_3 * y_3) \bmod q'$ = ((8 - 198)/(3 + 211 * 5 + 13 * 9 + 27 * 7) mod 281 = 128 and returns it to Alice. Alice checks the integrity of the components using the verification relation: $g_0^{c_0}(g_1^{x_1} g_2^{x_2} g_3^{x_3} h_0)^{r_0} \bmod p' == a_0$. In this case both a_0 and $3^{198} *$ $((243^{211}) * (541^{13}) * (498^{27}) * 27)^{128}$ mod 563 are equal to 368, so the verification relation holds. Alice finalizes $r_0' = r_0 + \gamma/\alpha \bmod q' = (128+2)/12 \bmod 281 = 245$. Alice stores the finalized credential $cred_i = (h, c_0', r_0') = (256, 204, 245)$ for the

next step in certifying a pseudonym with ICA_j.

4.5. Pseudonym Credentials

As in the augmented scheme example from [1], identity credential $cred_i$ is used as an input in obtaining pseudonym credential $cred_j$ from ICA_j, which is then used to obtain another pseudonym credential $cred_k$ from ICA_k.

In general, to issue a pseudonym credential, the ICA conducts a verification of the previous credential and the proposed pseudonym, and then issues a new credential on the pseudonym.

ICA_j Issuance of Pseudonym Credential. Alice presents her identity credential $cred_i$ in the showing protocol. ICA_j verifies this before issuing a pseudonym credential $cred_j$.

Alice sends $cred_i$ = (256, 204, 245), p_i = (34, 33) and a scalar s_j = 25 to ICA_j, as input to the pseudonym service, which will yield a new credential $cred_j$ = (440, 252, 63) on pseudonym point p_j = (199, 158). This process proceeds in two steps: first the submitted $cred_i$ is verified, after that, the new $cred_j$ is issued.

Verification of $cred_i$ consists of three checks: the integrity check, the proof of knowledge, and the coefficients check. The integrity of $cred_i$ is verified using the digital credentials verification relation. Given credential $\left(h, c_0', r_0'\right)$ verify that $c_0' == H\left(h \mid g_0^{c_0'} h^{r_0'}\right)$ Thus with $cred_i$ = (256, 204, 245), $g_0^{c'} h^{r_0'}$ = (3^{204}) (256^{245}) mod 563 = 99. This value corresponds to the h_2 calculated during the issuing protocol, so $H\left(h \mid g_0^{c_0'} h^{r_0'}\right) = 204$ which is equal to c_0', which satisfies the verification relation.

The next step is the proof of knowledge. For this example, let's assume random values w = 6 and c = 7. ICA_j creates a random challenge c = 7, and sends it to Alice who creates $proof_i = \left(c', x_1, x_2, x_3\right) = \left(30, 211, 13, 27\right)$ where the value for $c' = c/\alpha + w \bmod q = 7/12 + 6 \bmod 281 = 30$ and x_1, x_2, x_3 are Alice's attributes for $cred_i$. Alice sends $proof_i$ to ICA_j who verifies it by confirming the equality of $h^{c'} a$ mod p' = (256^{30})363 mod 563 = 485 with $g_1^{x_1} g_2^{x_2} g_3^{x_3} h_0^c$ = (243^{211*7})(541^{13*7})(76^{27*7})(27^7) mod 563 = 485; thus proof of knowledge has been confirmed.

In the final verification step, ICA_j verifies that $cred_i$ is on the coefficients of ξ_i. In this case correspondence is confirmed: $\hat{e}\left(p_i, p_i\right) = 13b + 211$, the coefficients of which correspond to the (x_1, x_2) received in the proof, above.

After this successful verification of $cred_i$, ICA_j issues a pseudonym credential to Alice. Let's assume the random values for the issue protocol to be α = 19, β = 26, γ = 32, w_0 = 18. ICA_j calculates new point $p_j = p_i * s_1$ = (34, 33) * 25 = (199, 158) and $\xi_j = \hat{e}(p_j, p_j) = \hat{e}$((199, 158), (199, 158)) = weil_pairing((199, 158), phi((199, 158))) = weil_pairing((199, 158), (44b + 19, 158)) = 151b + 29. Assuming $id(ICA_j)$ = 11, the credential is constructed on attributes (x_1 = 29, x_2 = 151, x_3 = 11). Alice calculates and retains $h = \left(g_1^{x_1} g_2^{x_2} g_3^{x_3} h_0\right)^{\alpha} \bmod p_1$ = ((289^{29})(326^{151})(349^{11})(470))^{19} mod 563 = 440.

Next ICA_j selects random w_0=18 and calculates $a_0 = g_0^{w_0} = 3^{18} \bmod 563 = 484$,

which it sends to Alice. Alice calculates $c_0' = H(h \mid h_2)$ where h = 197 and $h_2 = g_0^{\beta}\left(g_1^{x_1} g_2^{x_2} g_3^{x_3} h_0\right)^{\gamma} a_0 \bmod p'$ = $3^{26}((289^{29})(326^{151})(349^{11})(470))^{32}484 \bmod 563$ = 425, thus $c_0' = H(440 \mid 425) = 252$. Alice retains this c_0' and calculates $c_0 = (c_0' - \beta) \bmod q_1$ = (252 - 26) mod 281 = 226 which is sent to ICA_j. ICA_j produces signature data $r_0 = (w_0 - c_0) / (x_0 + x_1 y_1 + x_2 y_2 + x_3 y_3) \bmod q'$ = (18 - 226)/(13 + 29 * 15 + 151 * 19 + 11 * 17) mod 281 = 41, which is sent to Alice. Alice evaluates the issue-time verification relation, confirming that $g_0^{c_0}\left(g_1^{x_1} g_2^{x_2} g_3^{x_3} h_0\right)^{r_0} \bmod p'$ = $3^{226}((289^{29})(326^{151})(349^{11})(470))^{41}$ mod 563 = 484 equals the expected value of a_0 = 484. Alice proceeds to unblind the signature $r_0' = (r_0 + \gamma) / \alpha \bmod q'$ = (41 + 32)/19 mod 281 = 63 and stores the resulting credential $cred_j$ = (440, 252, 63).

ICA_k Issuance of Pseudonym Credential. Alice uses pseudonym credential $cred_j$ to obtain another pseudonym credential with ICA_k. Alice enters into the "show" protocol, supplying $cred_j$ = (440, 252, 63), $attr_j$ = (x_1 = 29, x_2 = 151, x_3 = 11), p_j = (199, 158) and s_2 = 17. ICA_k first checks the time verification relation for $cred_j$: $c_0' == H\left(h \mid g_0^{c_0'} h^{r_0'}\right)$. Here c_0' = 252, h = 440, and $g_0^{c_0'} h^{r_0'} \bmod p'$ = $3^{252}440^{63}$ mod 563 = 425, thus, the hash H(440|425), evaluates, as above, to 252 which is equal to c_0'. The verification relation is satisfied for $cred_j$.

The proof of knowledge follows the verification relation. ICA_k creates a random challenge, say c = 8, and sends it to Alice who creates $proof_j = (c', x_1, x_2, x_3)$ = (37, 29, 151, 11) where $c' = c/\alpha + w \bmod q$ = 8 /19 + 7 mod 281 = 37 and x_1, x_2, x_3 are the attributes $attr_j$. Alice sends $proof_j$ to ICA_k who verifies it by confirming that $h^{c'} a == g_1^{x_1} g_2^{x_2} g_3^{x_3} h_0^{c} \bmod p'$. Here, $h^{c'} a \bmod p'$ = $(440^{37})381$ mod 563 = 99 and $g_1^{x_1} g_2^{x_2} g_3^{x_3} h_0^{c}$ = $(289^{29*8})(326^{151*8})(349^{11*8})(470^{8})$ mod 563 = 99; thus the proof of knowledge verifies correctly.

For the last verification step, ICA_k performs a pairing and confirms that the coefficients of $\xi_j = \hat{e}(p_j, p_j) = \hat{e}((199,158),(199,158)) = 151b + 29$ correspond to (x_1 =29, x_2 = 151) of $cred_j$.

Once $cred_j$ has been verified, ICA_k proceeds to issue pseudonym credential $cred_k$ to Alice. For the issuance of $cred_k$ assume values of α = 7, β = 26, γ = 22, w_0 = 28 for the random values of the issue protocol.Alice calculates her new point p_k = p_j * s_2 = (199 , 158) * 17 = (99, 179) and ξ_k = $\hat{e}(p_k, p_k)$ = $\hat{e}$((99, 179), (99, 179)) = weil_pairing ((99 , 179), phi((99, 179))) = weil_pairing((99, 179), (269b + 97, 179)) = 197b + 190. Assuming $id(ICA_k)$ = 42, attributes $attr_k$ = (x_1 = 190, x_2 = 197, x_3 = 42) are used for issuance of $cred_k$.

Credential $cred_k = (h, c_0', r_0')$ is calculated as follows. First, Alice calculates and retains $h = \left(g_1^{x_1} g_2^{x_2} g_3^{x_3} h_0\right)^{\alpha} \bmod p_1$ = $((68^{190})(441^{197})(49^{42})(508))^{7}$ mod 563 = 92. Alice sends p_j = (199, 158) and s_2 = 17 to ICA_k, who calculates the same $attr_k$. Next ICA_k selects a random w_0, say the value 28, and calculates $a_0 = g_0^{w_0}$ = 3^{28} mod 563 = 147 and sends this a_0 to Alice. Alice calculates $c_0' = H(h \mid h_2)$ where h = 92 and $h_2 = g_0^{\beta}\left(g_1^{x_1} g_2^{x_2} g_3^{x_3} h_0\right)^{\gamma} a_0 \bmod p'$ = $3^{26}(68^{190} * 441^{197} * 49^{42} * 508)^{22} *$ 147 mod 563 = 513, so $c_0' = H(92 \mid 513) = 240$. Alice retains c_0' and calculates $c_0 = (c_0' - \beta) \bmod q_1 = (240 - 26) \bmod 281 = 214$ and sends it to ICA_k. Finally,

ICA_k produces signature data $r_0 = (w_0 - c_0)/(x_0 + x_1y_1 + x_2y_2 + x_3y_3) \bmod q'$ = ((28 - 214)/(23 + 190 * 25 + 197 * 29 + 42 * 27)) mod 281 = 211, which is sent to Alice. Alice evaluates the issuance-time verification relation, confirming that $g_0^{c_0}\left(g_1^{x_1}g_2^{x_2}g_3^{x_3}h_0\right)^{r_0} \bmod p'$ = $3^{214}((68^{190})(441^{197})(49^{42})(508))^{211}$ mod 563 = 147 which equals a_0, as expected. Alice proceeds to unblind the signature $r_0' = (r_0 + \gamma)/\alpha \bmod q_1$ = (211 + 22)/7 mod 281 = 234 and stores the resulting credential $cred_k$ = (92, 240, 234), along with the attributes and other parameters she used to obtain it.

4.6. Key Extraction

Alice sends credential $cred_k$ = (92, 240, 234) and pseudonymous point p_k = (99, 179) to the PKG with supporting attributes (190, 197, 42). The PKG checks the verification relation: $c_0' == H\left(h \mid g_0^{c_0'}h^{r_0'}\right)$, with h = 92, and $g_0^{c_0'}h^{r_0'}$ = 3^{240} * 92^{234} mod 563 = 513, thus $H(92|513) = SHA256(92|513)$ mod 281 = 3b11806f98cd81d368137bb211c63561bff95c219fc5b6649501708d0e14693b mod 281 = 240 which indeed equals c_0', so the verification relation holds.

The PKG's next step in checking $cred_k$ is the proof of knowledge. Proof of knowledge proceeds with the PKG generating random value w = 8 and Alice generating proof (c' = (9/7 + 8) mod 281 = 210, x_1 = 190, x_2 = 197, x_3 = 42). The PKG evaluates $(h^c)a = (92^{210})271$ mod 563 = 201 and confirms that it is equal to $\left(g_1^{x_1*c}\right)\left(g_2^{x_2*c}\right)\left(g_3^{x_3*c}\right)\left(h_0^c\right) = \left(68^{190*9}\right)\left(441^{197*9}\right)\left(49^{42*9}\right)\left(508^9\right)$ mod 563. Finally, the PKG confirms that the coefficients of $\xi_k = \hat{e}(p_k, p_k) = 197b + 190$ correspond to attributes (x_1, x_2).

Once $cred_k$ has been verified, the PKG creates the keying material by multiplying the pseudonymous point p_k = (99, 179) with PKG private master key t = 23 to obtain keying material $K = t * p_k$ = 23 * (99, 179) = (34, 248). This keying material is sent to Alice.

4.7. Finalization and Decryption

To finalize her decryption key, Alice applies the scalars she retained during pseudonym creation to the keying material received from the PKG to obtain K_A = $((s_1s_2)^{-1} \bmod q) * K$ = $((25 * 17)^{-1} \bmod 47)$ * (34, 248) = 24 * (34, 248) = (111, 206). Alice had received ciphertext $(u, v, w) = \left((263,167), 101001110100111011110011, 000001110101011000000001\right)$ from Bob. She proceeds to decrypt it using the decryption key K_A = (111, 206).

First, she computes $z = \hat{e}(K_A, u) = \hat{e}\left((111,206),(263,167)\right) = 21b + 60$. Then she derives $\sigma = v \oplus H_3(z)$ = 101001110100111011110011 $\oplus$ $H_3(21b + 60)$ = 101001110100111011110011 $\oplus$ 001100110011011000110001 = 100101000111100011000010. Using σ, she calculates the plaintext $M = w \oplus H_4(\sigma))$ = 000001110101011000000001 $\oplus$ 011001100011010001100010 = 011000010110001001100011. Alice confirms the verification relation: $u == r * G$ = 43 * (1, 132) which is equal to u = (263, 167), the first component in the ciphertext, so the ciphertext is confirmed to be valid. Encryption returns the string

value of M, which is "*abc*".

5. Discussion

The worked example of section 4 demonstrates small numbers for ease of calculation and for the sake of communication. This section discusses how to obtain parameters with more realistic security levels. BF-IBE specifies that an admissible mapping is required to achieve IND-ID-CCA security; this section also describes how the current implementation should be extended to achieve this.

5.1. Parameter Generation

The worked example uses small numbers for the sake of illustration. In an IBE deployment, the group parameters must be selected so that identity attacks on G_1 and forgery attacks on the digital credential are cryptographically hard. The parameter generation algorithm described in Section 5 can be modified to set target security sizes for F_p and $Z_{p'}$. For 128-bit security, we would seek $|q| \geq 254$ bits and $|p| \geq 3072$ bits [17]. If we want 128 bits of security, we will need p to be at least 1536 bits in length (so that the group size in F_{p^2} is at least 3072 bits long). One possible combination of parameters is shown in Table 2.

Table 2. Parameters for 128 bit security.

Parameter	Bits	Value
q	254	180849548655878188988365223120221389589733360668762679312917917098630926916 57
p	1536	150581587187683224368638579740541708674744213231908110946254909267298614891314771260509633794025966684340548026366823817955160383597564556262979063671810499208131435420800275716978025229082012410897784079781127610249755672633743612996898666717726284846619817012082787806973567339399948119685039700987765104422186176708790893012981539002832518027575046832719436167918402130680011807050477728535016309089979072355557479218606388186179852878087922201606603961450344 9
q'	1536	150581587187683224368638579740541708674744213231908110946254909267298614891314771260509633794025966684340548026366823817955160383597564556262979063671810499208131435420800275716978025229082012410897784079781127610249755672633743612996898666717726284846619817012082787806973567339399948119685039700987765104422186176708790893012981539002832518027575046832719436167918402130680011807050477728535016309089979072355557479218606388186179852878087922201606603961450344 9
p'	3072	362948670863965818887449974605726854711068850480972564930497711344548584389198822641609573332949845427071396045033518853972630478440806016764652849090696485317933186300183805974162455670937509189190091435274973825827102765916993330557387395253985736001885124136663840845429730650901285430861719860222549140270542120617535589728926275746464838040528384993151276620347601885185365759688170330817936406056642098832094117177224532881243911286262127234775186593610681550282603157461894045560539769290302985242767999711361368859371399591518977508568386418590316614012384366298194610503865828035138435926423974408138658869386958547336090472376286455631141398599655588122460118235939592683723920568205321376211727974225997978376610562789823902131127963021298736456827480419573488268763084591975675979599067437152855354163971318106081298711476201108369193785282139901839434772391404749761701860545988212060432971509954674457832340887 9

The required constraints hold on the above parameters. All numbers are prime, and all required relationships hold, namely that $q \mid p+1$, $p \equiv 2 \bmod 3$, $p \neq 3 \bmod 4$, $q^2 \nmid p+1$, and $q' \mid p'-1$.

We also note that other (*i.e.*, non-supersingular) elliptic curves may be used to obtain a security level of 128 bits (or higher) with a smaller value of p. For example, BLS12-381 and BLS12-440 curves have an embedding degree of $k = 12$, allowing a significant reduction in the size of p (which, of course, would lead to a corresponding increase in the efficiency of computations involving p); see [18]. However, these so-called *Type 3* curves use an asymmetric pairing ($\hat{e}: G_0 \times G_1 \to G_T$), rather than a symmetric pairing ($\hat{e}: G_1 \times G_1 \to G_T$), which would then require a modification to how pairings are used for pseudonym construction in PP-IBE.

5.2. IBE Topics

Security Model. Boneh and Franklin [3] [4] define IND-ID-CCA, a form of chosen ciphertext security applicable to IBE schemes. IND-ID-CCA defines adaptive security in which the attacker attempts to demonstrate an advantage at decrypting a ciphertext for a chosen identity. The Attacker is given a choice of which identity to attack, the option of knowing the decryption keys for a set of independent identities, and oracle access to the private key extraction algorithm and to the decryption algorithm.

Construction: Boneh and Franklin present a construction based on $p \equiv 2 \bmod 3$ with its accompanying distortion map. Our example proceeds with such a curve and is, as such, complimentary material for researchers in the area.

Distribution of Trust: A large amount of trust is placed in the PKG in an IBE deployment. All users obtain their decryption keys from the PKG using their public identity string. The PKG is thus able to derive private keys for all users. Adams' architectural approach wipes out this complete knowledge, displacing it across the community of ICAs instead. The initial ICA provides a verification of ownership of the identity string (for example, using a challenge and response email pattern). The subsequent ICA issues a pseudonym credential on the trust of the previous ICA's identity proofing mechanisms. Thus, the full trust that had previously been placed in the PKG is now spread out across the service providers:

- At the beginning of this chain of trust, the first ICA issues an identity credential, verifying Alice's ownership of the identity string, and then maps that string to a point (using a map-to-point function $H_1(.)$ or its stronger "admissible encoding" counterpart $L(.)$; see below).
- The subsequent ICAs in the chain each issue a pseudonym on the basis of their trust in the integrity and processes of the peers that precede them in the chain, the strength of the signatures, and the quality of the map-to-point function.

Admissible Encoding: Of the variations presented in [3] [4], only *FullIdent'* is IND-ID-CCA secure. The algorithms for *FullIdent* and *FullIdent'* are identical,

differing only differ only in terms of the definition of $H_1(.)$. *FullIdent'* tightens requirements on the mapping, such that the distribution is provably uniformly random. In *FullIdent'*, $p_1 = H_1(\text{id\_string})$ is replaced with $p_1 = L\left(H_1'(id\_string)\right)$ where $H_1'(.)$ is uniform string hashing and $L(.)$ is uniform point mapping. This is referred to as an "admissible encoding". The H_1 used in this implementation conforms to FullIdent requirements but has not (yet) been verified against FullIdent' requirements.

6. Performance

This section presents performance measures for the scenario of Figure 1 using the 128-bit security parameters presented in section 5.1.

The computing environment consists of SageMath version 9.6, installed on Ubuntu 20.04.5 LTS within the Windows Subsystem for Linux (GNU/Linux 5.10.16.3-microsoft-standard-WSL2 x86_64) on a Windows 10 Pro computer equipped with an IntelCore i7-1185G7, 3.00 GHz processor with 16.0 GB of RAM.

Table 3 presents performance measurements at 4 levels of drill-down: a) total demo execution time, b) time spent on initialization vs. protocol execution, c) breakdown by workflow step and, within each step, the breakdown per algorithm, d) within selected algorithms, a view at the primitive operations used. Full drill-down is shown for the issuance of credj, but is omitted from other steps for the sake of brevity.

The total time to run the demo is made up of two parts: initialization time, and protocol execution time. The actual protocol runs in 1.68 seconds. The full demo runs in 20 seconds, however data structure initialization accounts for 92% of this time, requiring over 18 seconds. Note that this initialization is a one-time pre-deployment cost to instantiate objects using prime moduli and curve parameters; once these are established, the PP-IBE system can be run for a set of users for an indefinite amount of time at sub-second speeds for each protocol step.

Figure 3 presents the relationship of protocol execution to system initialization and shows the execution times and proportions of the main steps in the scenario depicted in Figure 1.

Comparative Examination of Workflow Steps. When evaluated with the 128-bit security parameters, each protocol step exhibits sub-second performance. The three credential issuance steps are comparable, requiring between 304 and 462 ms. Also comparable are the encryption and decryption steps, requiring between 132 and 162 ms. These costs are dominated by the cost of the pairing. The issuance of $cred_j$ requires 2 pairing calculations, whereas the issuance of $cred_j$ and $cred_k$ requires three. Encryption and decryption each require one pairing. The key extraction step also requires a pairing for credential verification. Assuming upfront data structure initialization, key-pair establishment and the selection of generators requires 1.6 ms.

Primitive Usage. The PP-IBE protocol combines elliptic curve operations, pairings, and modular exponentiations. Figure 4 shows the breakdown of Step

Table 3. Performance measurements at 128 bit security.

Step	Performance (*ms*)
Demo	20381.8
Initialization	18699.5
Scenario	1682.2
1) *Setup*	1.6
2) Encrypt	132.5
3.1) *Total for* $cred_i$	304.0
Alice calculates ζ_i	127.8
ICA_i *issues* $cred_i$	176.1
ICA_i *calculates* ζ_i	132.0
Make h	8.5
Make a_0	0.02
Make c_0'	8.7
Make c_0	0.002
Make r_0	0.1
Make r_0'	0.02
Alice accepts $cred_i$	26.6
3.2) *Total for* $cred_j$	461.8
Alice calculates ζ_j	128.7
ICA_j *verifies* $cred_i$	160.3
Verifyintegrity	4.8
Verify ZKP	22.2
Verify Coefficients	133.1
ICA_j *issues* $cred_j$	172.54
ICA_j calculates ζ_j	129.06
Make h	8.4
Make a_0	0.02
Make c_0'	8.4
Make c_0	0.003
Make r_0	0.08
Make r_0'	0.02
Alice accepts $cred_j$	26.49
3.3) *Total for* $cred_k$	462.0
Alice calculates ζ_k	129.2
ICA_k *verifies* $cred_j$	159.3
ICA_k *issues* $cred_k$	147.06
Alice accepts $cred_k$	26.28
4) Total for PKG	158.2
Verify $cred_k$	157.7
Extract key	0.5
5) Total for Finalize and Decrypt:	162.0
Finalize key:	3.3
Decrypt:	158.6

Continued

Scalar Multiplication:	0.34
Modular exponentiation:	0.06
Mapto Point:	3.14
Weil Pairing:	125.45

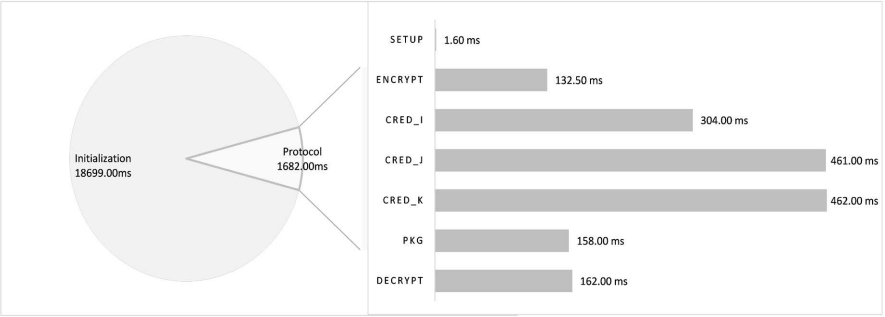

Figure 3. Protocol execution (Setup, Encrypt, Credential Certifications, Key Extraction and Decrypt) takes approximately 1.68 seconds. Initialization activities for group and curve data structures require 18.7 seconds. Each protocol step exhibits sub-second performance. The encrypt and decrypt steps have comparable performance, as do the three credential issuance steps.

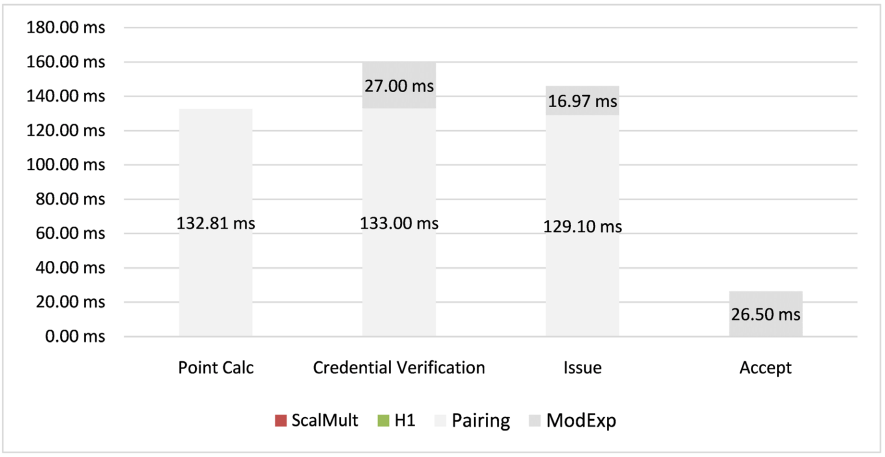

Figure 4. Looking at Step 3.2, the cost of pairing operations dominatesissuance of $cred_j$. The total cost of this step is 461.8 ms; 85% of that time is spent in pairings.

3.2, the interactive protocol between Alice and ICA_J for the issuance of $cred_j$, and provides a view of the usage of the primitive operations and how the costs of these impact protocol performance.

The total cost of step 3.2, the issuance of cred_j, is made up of four parts. Alice's calculation of the pseudonym ζ_j, the ICA verification of $cred_j$, the issuance of $cred_j$, and Alice's acceptance of $cred_j$. The first three parts each require a pairing calculation, at approximately 130 ms. The modular exponentiations require a fraction of that time. Looking at credential issuance and acceptance in steps 3 and 4, over 20 modular exponentiations are required, at a total of 43.5 ms, these

making up only 25% of the total 172.54 ms required.

7. Conclusions

In a Boneh-Franklin Identity Based Encryption (BF-IBE) [3] [4], the Private Key Generator has significant power, with the ability to derive the decryption keys of the community of users. The community must trust, therefore, that this PKG will not be malicious.

The recently-introduced Privacy Preserving Identity-based Encryption ("PP-IBE") [1] [2] removes this complete knowledge of the PKG, and distributes it across the PKG and a community of Intermediate Certification Authorities (ICA) who grant identity and pseudonym credentials. In PP-IBE, the generation of decryption keys becomes a collaborative responsibility between the ciphertext recipient and the PKG.

This paper provides an elaboration of the recent privacy preserving IBE proposed by Adams. We offer a construction and a worked example based on torsion groups within supersingular elliptic curves over F_p in which $p = 2 \bmod 3$, along with their accompanying distortion map. We make extensions to Adams' algorithms, including a parameter generator, an added digital credential base to support our approach of carrying the Weil or Tate pairing coefficients, and a reuse opportunity between the issuance of identity credentials and pseudonym credentials. We offer a software implementation and a worked numeric example that may be useful to researchers and to students new to this area. We also offer selected discussions into representative-sized parameters for privacy and security properties in an open-world setting.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Adams, C. (2022) Improving User Privacy in Identity-Based Encryption Environments. *Cryptography*, **6**, Article No. 55. https://doi.org/10.3390/cryptography6040055

[2] 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

[3] Boneh, D. and Franklin, M. (2001) Identity-Abased Encryption from the Weil Pairing. In: Kilian, J., Ed., *Advances in Cryptology—CRYPTO* 2001. *CRYPTO* 2001. *Lecture Notes in Computer Science*, Vol. 2139, Springer, Berlin, 213-229. https://doi.org/10.1007/3-540-44647-8_13

[4] 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

[5] Brands, S. (2002) A Technical Overview of Digital Credentials.

https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=9c2f1b143a9f07e8644aa72d57168638b7f469f4

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

[7] SageMath (Free Opensource Mathematics Software System). https://www.sagemath.org/

[8] Silverman, J.H. (2009) The Arithmetic of Elliptic Curves. In: *Graduate Texts in Mathematics*, Vol. 106, Springer, New York. https://doi.org/10.1007/978-0-387-09494-6

[9] Menezes, A., Vanstone, S. and Okamoto, T. (1991) Reducing Elliptic Curve Logarithms to Logarithms in a Finite Field. *Proceedings of the* 23*rd annual ACM symposium on Theory of Computing*, New Orleans, 5-8 May 1991, 80-89. https://doi.org/10.1145/103418.103434

[10] Galbraith, S.D., Paterson, K.G. and Smart, N.P. (2008) Pairings for Cryptographers. *Discrete Applied Mathematics*, **156**, 3113-3121. https://doi.org/10.1016/j.dam.2007.12.010

[11] Freeman, D., Scott, M. and Teske, E. (2010) A Taxonomy of Pairing-Friendly Elliptic Curves. *Journal of Cryptology*, **23**, 224-280. https://doi.org/10.1007/s00145-009-9048-z

[12] Lynn, B. (2007) On the Implementation of Pairing-Based Cryptosystems. Stanford University, Stanford.

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

[14] Page, D., Smart, N.P. and Vercauteren, F. (2006) A Comparison of MNT Curves and Supersingular Curves. *Applicable Algebra in Engineering, Communication and Computing*, **17**, 379-392. https://doi.org/10.1007/s00200-006-0017-6

[15] Heath, L.S. and Loehr, N.A. (2004) New Algorithms for Generating Conway Polynomials over Finite Fields. *Journal of Symbolic Computation*, **38**, 1003-1024. https://doi.org/10.1016/j.jsc.2004.03.002

[16] Lübeck, F. (2023) Standard Generators of Finite Fields and Their Cyclic Subgroups. *Journal of Symbolic Computation*, **117**, 51-67. https://doi.org/10.1016/j.jsc.2022.11.001

[17] Barker, E., Barker, W., Burr, W., Polk, W. and Smid, M. (2007) NIST Special Publication 800-57. NIST Special Publication 800-57. National Institute of Standards and Technology, Gaithersburg, 1-142.

[18] Kumar, M. and Chand, S. (2022) Pairing-Friendly Elliptic Curves: Revisited Taxonomy, Attacks and Security Concern. ArXiv Preprint ArXiv: 2212.01855.

Appendix—Notation

Symbol	Description
$P = (q, p, q', p')$	Parameter specification for an instantiation of PP-IBE
r	The order of the r-torsion group G.
q	Prime integer specifying the number of elements of G_1 and G_T. The security of the scheme in PP-IBE depends on q. In our construction, $q = r$, because G_1 is chosen to be the torsion group of order r
$\mathcal{O}$	The point at infinity of an elliptic curve
p	Prime integer. Base of the prime field F_p from which points in $E(F_p)$ are drawn
p'	Prime number, the base of $Z_{p'}$ the prime field for digital credentials. p' is selected using q' such that $q' \mid (p'-1)$
$Z_{p'}$	The prime field within which calculations for digital credentials and their signatures are performed.
$Z_{q'}$	Prime field, source of the attributes and exponents for digital credentials
$\#E(F_p)$	The order of the base elliptic curve. E is specified in this document such that $q \mid o$.
$E(F_p)[q]$	The q-torsion group of curve E. In our construction $G_1 = E(F_p)[q]$.
k	Embedding degree k is the smallest positive integer k such that $\#G_1 \mid \#E_k + 1$. In our implementation the Tate pairing requires k as an argument.
F_p	Prime field upon which the base elliptic curve $E(F_p)$ is drawn. F_p is the set of integers $[0, p\text{-}1]$.
$E(F_p)$	The base elliptic curve. In this paper, $E(F_p): y^2 = x^3 + 1, \quad x, y \in F_p$
G_1	The set of points, a subset of $E(F_p)$, that forms the domain for the bilinear pairing function. Our construction presents a symmetric pairing in which G_1 is a q-torsion subgroup of $E(F_p)$. Group G_1 is the source of identity points for the protocol; as such, its size is a privacy and security parameter.
$E(F_{p^2})[q]$	The extension of G_1 into F_{p^2}. For a point $p_1 \in E(F_p)[q]$, $\varphi(p_1) \in E(F_{p^2})[q]$
$p_2 = \varphi(p_1)$	Distortion map which transforms a point from the torsion group $\mathbb{G}_1$ in the base curve $E(F_p)$ to a point in the torsion group Gx in the elliptic curve on the extension field $E(F_{p^2})$. In this paper we use $\varphi(x, y) = (\zeta x, y)$
$E(F_{p^2})$	The elliptic curve on the extension field. The set of pairs drawn from elements of the polynomial ring which satisfy the elliptic curve characteristic equation. In this paper, $E(F_{p^2}) = \{(x, y)\}$ x, y in F_{p^2}, s.t. $y^2 = x^3 + 1$.

Continued

F_{p^2}	Field extension of F_p, elements of which have the form $c_1x + c_0$ with coefficients $c_1, c_0 \in F_p$. In this paper $G_T \subset F_{p^2}$ and $E\left(F_{p^2}\right) \subset F_{p^2} \times F_{p^2}$.
$\hat{e}: G_1 \times G_1 \to G_T$	Custom pairing function $\hat{e}$ allowing two points from G_1 (possibly dependent) to be paired in a non-degenerate manner to a point in G_T. Our construction implements $\hat{e}$ in terms of e and φ
$f: G_1 \times G_1 \to G_T$	A well-known pairing function, such as the Weil or the Tate pairing.
G	Source Group for the pairing. In our construction the source group G is the r-torsion, a subset of the points in Ep
G_t	Target Group. In our construction, the target group is F_{p^2}, the extension field
$Z_{p'}$	The prime field of integers modulo p'. used for calculation of digital credentials.
$Z_{p'}^*$	The integers in $Z_{p'}$ relatively prime to p'
$Z_{q'}$	The prime field of integers mod q'. The values used for attributes and exponents within the digital credential are in $Z_{q'}$
r	The order of the r-torsion group G.
q	Prime integer specifying the number of elements of G_1 and G_T. The security of the scheme in PP-IBE depends on q. In our construction, $q = r$, because G_1 is chosen to be the torsion group of order r
$\mathcal{O}$	The point at infinity of an elliptic curve
p	Prime integer. Base of the prime field F_p from which points in $E(F_p)$ are drawn
p'	Prime number, the base of $Z_{p'}$ the prime field for digital credentials. p' is selected using q' such that $q' \mid (p' - 1)$
$Z_{p'}$	The prime field within which calculations for digital credentials and their signatures are performed.
$Z_{q'}$	Prime field, source of the attributes and exponents for digital credentials
$\#E(F_p)$	The order of the base elliptic curve. E is specified in this document such that $q \mid o$.
$E\left(F_p\right)[q]$	The q-torsion group of curve E. In our construction $G_1 = E\left(F_p\right)[q]$.
k	Embedding degree k is the smallest positive integer k such that $\#G_1 \mid \#E_k + 1$. In our implementation the Tate pairing requires k as an argument.
F_p	Prime field upon which the base elliptic curve $E(F_p)$ is drawn. F_p is the set of integers $[0, p - 1]$.
$E(F_p)$	The base elliptic curve. In this paper, $E\left(F_p\right): y^2 = x^3 + 1, \quad x, y \in F_p$

Continued

G_1	The set of points, a subset of $E(F_p)$, that forms the domain for the bilinear pairing function. Our construction presents a symmetric pairing in which G_1 is a q-torsion subgroup of $E(F_p)$. Group G_1 is the source of identity points for the protocol; as such, its size is a privacy and security parameter.
$E\left(F_{p^2}\right)[q]$	The extension of G_1 into F_{p^2}. For a point $p_1 \in E\left(F_p\right)[q]$, $\varphi(p_1) \in E\left(F_{p^2}\right)[q]$
$p_2 = \varphi(p_1)$	Distortion map which transforms a point from the torsion group $\mathbb{G}_1$ in the base curve $E(F_p)$ to a point in the torsion group Gx in the elliptic curve on the extension field $E\left(F_{p^2}\right)$. In this paper we use $\varphi(x, y) = (\zeta x, y)$
G	Source Group for the pairing. In our construction the source group G is the r-torsion, a subset of the points in
$Z_{p'}$	The prime field of integers modulo p'. used for calculation of digital credentials.
$Z_{p'}^*$	The integers in $Z_{p'}$ relatively prime to p'
$Z_{q'}$	The prime field of integers mod q'. The values used for attributes and exponents within the digital credential are in $Z_{q'}$