Multiple Trusted Authorities in Identifier Based Cryptography from Pairings on Elliptic Curves

Liqun Chen, Keith Harrison
Trusted Systems Laboratory
HP Laboratories Bristol
HPL-2003-48
March 19th , 2003*

E-mail: {liqun.chen, keith.harrison}@hp.com

identifier based cryptography, multiple trusted authority, pairings

We investigate a number of issues related to a key generation service in identifier based cryptographic technology. In particular, we focus on how to make this service more trustworthy. Our solution is the use of multiple trusted authorities in identifier based cryptography (MTAIBC), where these trusted authorities work together to issue a private key based on one or more given identifiers. However, to do this, these trusted authorities do not need to communicate with each other to establish a shared secret; and the users can freely choose a number of trusted authorities for their application purposes, which is not dependent on whether or not those trusted authorities have an agreement to work together. In the document, we give seven algorithms, each for a specific construct of the trusted authorities, and we also introduce a few examples of possible applications that can be implemented by using these algorithms.

* Internal Accession Date Only
Approved for External Publication

Contents

1 Overview

1.1 Introduction

Identifier Based Cryptography (IBC) is a new asymmetric cryptographic technology, which includes the techniques of Identifier Based Encryption (IBE), Identifier Based Signatures (IBS), Identifier Based Authenticated Key Agreement etc.. Compared with a traditional asymmetric cryptographic system based on a Public Key Infrastructure (PKI) [ISO/IEC 9594-8], this technology has an interesting property that verifying the validation of a user's public key and using the public key, such as encryption, signing and authentication, are handled in a single algorithm. Whilst in PKI verifying a public key and using the public key are simply two separate algorithms.

Let us consider a common situation – for the purpose of verifying the validation of a user's public key, we require a trusted third party, which owns a public/private key pair and the public key is publicly accessible.

In a PKI based system, this trusted third party is a Certificate Authority (CA) who issues a certificate on a user's public key after authenticating the user. The certificate is normally a signature on the user's public key signed under the CA's private signing key. To use the user's public key with validation checking, one needs first to verify validation of the user's public key by verifying the CA's signature in the certificate, and second to do encryption, signature verification or other cryptographic process with the user's public key.

In an IBC based system, this trusted third party is a Trusted Authority (TA) who issues a user's private key, after authenticating the corresponding public key of the user, in the way that the user's private key is combined with the TA's private key. As a result, the TA's public key is involved in every use of the user's public key and private key. The combination between the user's public key and the TA's public key is guaranteed since these two keys are both involved in every related cryptographic algorithm.

This can extend to a more interesting property for asymmetric cryptosystem, i.e., it results in building a non-pre-interactive communication between an encryptor and a decryptor. For example, If Alice wants to send a secure email to Bob at bob@hp.com she simply encrypts her message using the public key string bob@hp.com. There is no need for Alice to obtain Bob's public key certificate. For the purpose of making sure that only Bob is able to decrypt the message, Alice includes a public key of a TA within the encryption. After receiving the encrypted email Bob authenticates himself to the TA by convincing of the ownership of the email address to obtain his private key. Bob can then read this email. Note that unlike the existing PKI based secure email infrastructure, Alice can send encrypted email to Bob even if Bob has not yet got his private key and his public key certificate.

Note that in an ordinary IBC system we cannot escape the property of key escrow – it is obvious that the TA knows the user's private key. A potential problem from this property is that the TA may impersonate a user in the system because the TA is always able to do so. In an ordinary PKI system we have the same problem in fact. A CA can generate a key pair, and (falsely) certify that the public key belongs to a specific user. The CA can then impersonate the user to any other users. In both IBC and PKI we therefore always assume that the trusted authority (TA or CA) will not impersonate users. However, in PKI, the problem can be solved if we add an extra process -

which is actually recommended by many applications - that possession of the user's private key is verified every time when the user uses the key to communicate with others. Unfortunately, we cannot offer the same solution for IBC because key escrow is inherent in IBC. As a result an identifier-based signature is not able to provide the property of non-repudiation (even if the ownership of a private signing key has been proved by a user), since there is always more than one entity knowing the private signing key.

Our goal in this document is to keep the "natural" combination between the TA's public key and the user's public key, but reduce a single TA's power, and make the trusted authority service for IBC more trustworthy. Our solution makes use of Multiple Trusted Authorities in Identifier Based Cryptography (MTAIBC).

1.2 Brief History of IBC

1984. Shamir [Sh84] first introduced the concept of identity-based cryptography[1], including identity-based encryption and signatures. Shamir proposed an identity-based signature scheme based on the RSA assumption but left implementation of IBE as an open problem.

1986. Desmedt and Quisquater [DQ86] made a proposal to implement identity-based cryptosystems by using tamper-proof hardware devices.

1986. Okamoto [Ok86] presented an identity-based key agreement scheme and Tanaka and Okamoto slightly modified this in [TO91].

1987. Tanaka [Ta88] proposed an identity-based public key distribution scheme. This system becomes insecure if a number of users combined their private keys since this information allows them to jointly recover the private key of the key generation authority.

1987. Okamoto [Ok87] described a key distribution system based on identification information. This system was not truly directory-less since the user's identity is only part of the key generation process and a public directory is still required for offline communication between users.

1990. Girault and Pailles [GP90] developed an identity-based system, which can be used for non-interactive key agreement schemes.

1991. Maurer and Yacobi [MY91] proposed an identity-based public key distribution scheme. The system proposed, however, required considerable computational effort on the part of the trusted key generation authority.

1997. Vanstone and Zuccherato [VZ97] proposed an identity-based key distribution scheme. This system failed to provide adequate security on a number of grounds, since the composite modulus proposed can be factored easily and the computational effort required on the part of the trusted key generation authority was not much less than that required to break the system by an unauthorised user.

1998. ISO/IEC 14888-2 was published [ISO/IEC 14888-2], which includes two IBS schemes based on works from [GQ88] and [GS94].

1999. ISO/IEC 11770-3 was published [ISO/IEC 11770-3], which includes an identity-based key agreement scheme based on works from [Ok86], [TO91] and [GP90].

[1] Shamir and many other researchers call it identity-based cryptography. The authors of the document prefer to call it identifier-based cryptography because the information used as user's public key in many applications is actually not only identity but also other terms and conditions.

2000. Sakai, Ohgishi and Kasahara [SOK00] proposed an identity based signature algorithm using Weil pairing, where 3 Weil pairings are used for signature verification and one of them may be precomputed.

2001. Two identity based encryption algorithms were proposed, one by Cocks [Co01] based on the quadratic residuosity assumption and another by Boneh and Franklin [BF01a] based on the Weil pairing. Some later researches described that using a variant based on the Tate pairing is more efficient [Ga01].

2001-2002. A number of identity-based signature schemes were published, including

- Galbraith's scheme [Ga02], where four Weil pairings are needed for signature verification.
- Hess's scheme [He02], where one Tate pairing, that may be precomputed, is used for signature generation and two Tate pairings, where one of them may be precomputed, are used for signature verification.
- Modified Hess's scheme [MCH03], which is modified by Chen.
- Paterson's scheme [Pa02], where also includes one non-pre-computable pairing and two pre-computable pairings.
- Cha's and Cheon's scheme [CC02], where one Weil pairing is used for signing and another is used for verification.
- Soldera's scheme [So02], where two Tate pairings are used for signature verification.

2002. Smart [Sm02] proposed an identity-based authenticated key agreement protocol. After that, Chen and Kudla [CK02] modified Smart's scheme to make it more efficient and, more interesting, to make it with the new property of TA forward security, which they define to mean that the compromise of the TA's private key will not compromise previously established session keys.

1.3 Motivation of This Work

In this subsection, we would like to explain the reason why MTAIBC needs more research.

As it has been mentioned above, an existing ordinary IBC scheme (for examples, [BF01a], [Co01], [He02], etc.) allows a single TA to issue an identifier-based private key for a user. Therefore, this TA can masquerade any of his users if he wants to. Even if he does not actually abuse any user's key, a user may still be able to deny some use of the key, e.g. a signer can deny an identifier-based signature by claiming to no longer trust the TA who has issued the signing key. It makes non-repudiation very difficult.

To solve the problem of key escrow in IBC, a number of researchers have proposed some solutions of splitting the authority into two or more co-operating parties. For example, the author of the QR-based IBE method, Cocks in [Co01] proposed a secret sharing protocol, in which more than one TA can establish a shared public parameter, which is the product of two large primes as private parameters. The protocol ensures that neither of the TAs knows the values of the private parameters. Boneh and Franklin proposed a similar scheme in [BF01b]. The authors of pairing-based IBE method, Boneh and Franklin in [BF01a] proposed another secret sharing protocol, in which n TAs each has a share of a secret master key, and by using a t-out-of-n threshold scheme [Sh79]: any t TAs are able to recover the master key, but any $t - 1$ TAs or less cannot get any information about the master key.

Both of the above solutions stopped a single TA escrowing a user's private key by letting a group of TAs share a secret. However, we can argue that letting a group of TAs share a secret has the following disadvantages:

1. It is not flexible. If the group's construction changes, e.g. any group member leaves or a new member joins, a new-shared secret has to be generated. In an IBE application, an encryptor may want to select which set of TAs they will use in encrypting any given message. So the construction of the group of TAs needs to be changed from case to case. There have been a number of papers talking about how to make the shared secret reusable (e.g., [Pi96] and [CGMW97]), but none of these schemes is efficient.

2. One authority may be involved in many different groups providing different services. For each group construction, he needs to make an individual contribution to an individual shared secret. Users may find it difficult to trust this authority since he has many different "faces" (public keys). On the other hand, users have to access many different public keys from a single TA, each public key for a given application.

3. More importantly, in some real-life applications, it may be impossible to organise those trusted authorities to run such a secret sharing protocol, since, for various reasons, some of them may not be interested in co-operating with the others, and also since users may not want these TAs to communicate each other before offering their services to the users.

4. The solutions proposed are organized by the TAs themselves. They do not permit the encryptor to choose their own TAs.

Based on the above consideration, it is required to have an easier and more flexible model of MTAIBC.

1.4 Players and Their Requirements

In a general MTAIBC application, there are three kinds of entities involved: Alice, Bob, and a set of TAs. Alice is either an encryptor in IBE or a verifier in IBS. Bob is either a decryptor in IBE or a signer in IBS.

What do these players require for MTAIBC?

- Alice. She may want to select a subset of TAs for a given message to be encrypted; and she may also want to control contents and freshness of Bob's public encryption key.

- Bob. He may want to select a subset of TAs for a give message to be signed; and he may also want to control contents and freshness of his own keys.

  For meeting both Alice's and Bob's requirements, they may negotiate an agreement of a subset of TAs they will use in a given application.

- TAs. Each TA may not want to change their public key for different users and applications. TAs may not want to communicate with each other to establish a shared secret. There is no interaction needed between the TAs.

To consider the above requirements of entities, in this document, we propose a robust scheme based on the existing IBE and IBS methods to provide multiple trusted authority services. This scheme has the following properties:

1. The TA doesn't have to change its public key for any group constructions.
2. TAs do not need to communicate with each other to establish a shared secret.
3. The users can freely choose a number of trusted authorities for their application purposes, which is not dependent on whether or not those trusted authorities have an agreement to work together.

2 Symbols and Definitions

2.1 Symbols

This document makes use of the following symbols and notations.

E A elliptic curve such as that defined by $y^2 = x^3 + 1$ over $\mathbb{F}_p$.

$\mathbb{F}_p$ The Galois field of integers modulo p, comprising the integers $[0, p - 1]$.

G_1 A group of points on an elliptic curve with an additive notation, which has prime order q and in which the discrete logarithm problem is believed to be hard.

G_2 A subgroup of the multiplicative group of a finite field with a multiplicative notation, which has prime order q and in which the discrete logarithm problem is believed to be hard.

H_1 The operation of a hash-function, which takes a string as input and outputs a point of G_1, i.e. if A is a string of $\{0, 1\}^*$ then $H_1(A)$ denotes the point mapped from a hash code of m, i.e., $H_1 : \{0, 1\}^* \rightarrow G_1$.

H_2 The operation of a hash-function, which takes an element of G_2 as input and outputs a string of $\{0, 1\}^*$, i.e. if A is an element of G_2 then $H_2(A)$ denotes a hash code of A, i.e., $H_2 : G_2 \rightarrow \{0, 1\}^*$.

ID A string of $\{0, 1\}^*$, which is an identifier of a user.

ID_j A string of $\{0, 1\}^*$, which is the j^{th} identifier of a user.

p The size of the field $\mathbb{F}_p$, which is a prime such that $p = 2 \bmod 3$ and $p = 6q - 1$ for some prime $q > 3$.

P An arbitrary point of G_1.

P_i An arbitrary point of G_1.

$\mathbb{p}$ A pairing with entries chosen from two points of an elliptic curve, i.e., if A and B are such points, then $\mathbb{p}(A, B)$ is the data string by operation of either the Weil pairing or Tate pairing of A and B.

q A prime number, which is order of G_1 and G_2.

Q A point of G_1, which is mapped from ID, i.e., $Q = H_1(\text{ID})$.

Q_i, Q_j Points of G_1, which are mapped from ID_i and ID_j respectively, i.e., $Q_i = H_1(\text{ID}_i)$ and $Q_j = H_1(\text{ID}_j)$.

R A point of G_1 used as entity TA's public parameter.

R_i A point of G_1 used as the i^{th} TA's public parameter.

s An arbitrary data string used as a private master key of the TA.

s_i An arbitrary data string used as a private master key of the i^{th} TA.

S A point of G_1 used as a private key corresponding to Q issued by a TA, i.e., $S = sQ$.

S_i A point of G_1 used as a private key corresponding to Q issued by a TA, i.e., either $S_i = s_iQ$, $S_i = sQ_i$ or $S_i = s_iQ_i$.

S_{ij} A point of G_1 used as a private key corresponding to Q_i issued by a TA, i.e., $S_{ij} = s_iQ_j$.

$\oplus$ The operation of XOR.

|| Concatenation of two data strings.

2.2 Definitions

For the purposes of this document, the following definitions apply.

2.2.1 Weil Pairing and Tate Pairing

A pairing is a computable bilinear map between two groups G_1 and G_2. Two pairings have been studied for cryptographic use. They are the (modified) Weil pairing $\hat{e}, : G_1 \times G_1 \rightarrow G_2$ [MOV93, Si94, BF01a], and the (modified) Tate pairing $t : G_1 \times G_1 \rightarrow G_2$ [FMR99, Ga01].

For the purposes of this document, we let $\mathbb{p}$ denote a general bilinear map, i.e.,

$$\mathbb{p}: G_1 \times G_1 \rightarrow G_2,$$

which can be either the modified Weil pairing or the modified Tate pairing, and which has the following three properties:

- Bilinear: If $P, P_1, P_2, Q, Q_1, Q_2 \in G_1$ and $a \in \mathbb{Z}^*_q$,
  $\mathbb{p}(P_1 + P_2, Q) = \mathbb{p}(P_1, Q)\ \mathbb{p}(P_2, Q)$,
  $\mathbb{p}(P, Q_1 + Q_2) = \mathbb{p}(P, Q_1)\ \mathbb{p}(P, Q_2)$, and
  $\mathbb{p}(aP, Q) = \mathbb{p}(P, aQ) = \mathbb{p}(P, Q)^a$.
- In general $\mathbb{p}(P, P) \neq 1$.
- Computable: If $P, Q \in G_1$, one can compute $\mathbb{p}(P, Q)$ in polynomial time.

2.2.2 Standard Public/Private Key Pair

A standard public/private key pair is a pair (R, s) where $R \in G_1$ and $s \in \mathbb{Z}^*_q$ with

$$R = sP$$

for some given fixed point $P \in G_1$.

2.2.3 Identifier Based Public/Private Key Pair

An identifier based key pair is a pair (Q, S) where $Q, S \in G_1$ and there is some trust authority (TA) with a standard public/private key pair given by (R, s), such that the key pair of the trust authority and the key pair of the identifier are linked via

$$S = sQ \text{ and } Q = H_1(ID);$$

where ID is the identifier.

2.2.4 Trust Authorities

A Trust Authority (TA) in this document is defined to have the following properties:

- Having a standard public/private key pair (R, s), where the public key $R \in G_1$ and the private key $s \in \mathbb{Z}^*_q$, with $R = sP$ and P is a public point.
- Having the ability to convert an identifier string ID to an identifier based public/private key pair (Q, S) with

$$S = sQ \text{ and } Q = H_1(ID).$$

3 Concepts of IBC

In Identifier Based Cryptography (IBC) a user's public key is derived directly from the user's identifier string and associated with a TA's public parameters, which ensures the user's public key naturally bound with the TA's public key, and which allows the TA (not the user) to be able to issue the corresponding private key of the user. This natural key combination replaces a certificate combination in a traditional PKI based system, which makes the IBC system very attractive.

3.1 An IBE scheme

For the purpose of this document we sketch the basic principles of the scheme of Boneh and Franklin [BF01a] as follows.

The scheme of Boneh and Franklin allows the holder of the private part of an identifier based key pair to decrypt a message sent to her under the public part. We present only the simple scheme, which is only ID-OWE, for an ID-CCA scheme one applies the Fujisaki-Okamoto transformation [FO99, BF01a].

There are three players involved in the scheme: Alice - an encryptor, Bob - a decryptor and TA – an off-line trusted authority. TA's public key is $(P, R = sP \in G_1)$ and TA's private key is $s \in \mathbb{Z}^*_q$. When Bob registers with TA, who issues a private key $S = sQ \in G_1$ for him, where $Q = H_1(\text{ID}) \in G_1$, and ID is Bob's identifier string.

Let m denote the message to be encrypted.

For encryption, Alice computes $U = rP$ where r is a random element of $\mathbb{Z}^*_q$, and

$$V = m \oplus H_2(\mathbb{p}(R, Q)^r),$$

then outputs the ciphertext (U, V).

For decryption, Bob computes

$$V \oplus H_2(\mathbb{p}(U, S)) = V \oplus H_2(\mathbb{p}(rP, sQ)) = V \oplus H_2(\mathbb{p}(R, Q)^r) = m.$$

3.2 An IBS scheme

For the purposes of this document, we sketch the modified Hess scheme. This scheme was invented by Hess [He02] and modified by Chen.

There are three players involved in the scheme: Alice– a signer, Bob - a verifier and TA– an off-line trusted authority. TA's public key is $(P, R = sP \in G_1)$ and TA's private key is $s \in \mathbb{Z}^*_q$. When Alice registers with TA, who issues a private key $S = sQ \in G_1$ to her, where $Q = H_1(\text{ID}) \in G_1$, and ID is Alice's identifier string.

Let m denote the message to be signed.

For signing, Alice computes

$$r = \mathbb{p}(S, P)^k,$$

where k is a random element of $\mathbb{Z}^*_q$, applies the hash function H_2 to $m\|r$ (concatenation of m and r) to obtain

$$h = H_2(m\|r).$$

Then she computes

$$U = (k\text{-}h)S,$$

and outputs (U, h) as the signature on the message m.

For verification, Bob computes

$$r' = \mathbb{p}(U, P)\mathbb{p}(Q, R)^h.$$

He accepts the signature if and only if $h = H_2(m||r')$.

3.3 Hierarchies of TAs

By a hierarchy of multiple TAs, we mean architecture of linking a number of TAs, in which these TAs are on different levels. On the top level, there is one or more root TA, each of which has a standard asymmetric key pair. On the other levels, there is one or more TA, each of which has an IBC key pair. A TA on the immediately above level issues the private part of this key pair.

There have recently been a number of different solutions of hierarchy IBC, for examples, [HL02], [CHMSS02], and [GS02].

In this document, we focus on only one level TAs. But the solution described in this document can be extended to multiple levels.

3.4 Calculus of TAs

There are a number of different ways of a group of TAs working together to offer a trust authority service for IBC.

About ten years ago, Desmedt [De87] and Boyd [Bo88] [Bo89] presented the notion of group oriented cryptography, namely key calculus.

The key calculus MTAIBC includes three different operations. They are

- Addition, in which a group of TAs have to work together in order to make an additional operation;
- Or, in which any TA from a group of TAs can use their single key to make the operation; and
- (t, n) threshold, in which any t members from a group of TAs with totally n members can work together to make the operation.

More detailed information of key calculus with MTAIBC can be found in [CHSS02]. In this document, we will focus on the operation of Addition. Again, the solution we get here can be extended to the other operations.

4 Construction of Multiple TAs

4.1 Scope

In this section we will describe how to achieve the pairing based IBC algorithms with multiple trusted authorities and/or multiple identifiers.

To setup a system with multiple TAs, i.e., to setup keys for those TAs involved, there are the following different methods:

1. Those TAs choose their system parameters separately, i.e., each TA sets up their own groups G_1 and G_2 with order q, their own point $P \in G_1$ and their own master private key $s \in \mathbb{Z}^*_q$.
2. Those TAs take the same system parameters and then generate their own master private keys. More specifically, a group of TAs make use of an elliptic curve, which creates groups G_1 and G_2, and a single point P, which may be chosen by a standard body. Alternatively, those TAs partially take the same system parameters; for example, a group of TAs make use of an elliptic curve that may be chosen by a standard body, but they each chooses their own point P.
3. Those TAs share some secret as a system master key, as mentioned earlier, these TAs may run a protocol to get their share.

In any of the above methods, the encryption can be trivially done using an onion form of encryption where each encryption is applied in turn to obtain the ciphertext. Note that it is not a very elegant solution and it also implies that the decryptor needs to apply the necessary decryptions in the same order that the encryptor applied the encryptions. This is also suitable for signatures.

However, when the first method is required, we may have to recommend that it can only be done using an onion form where, for example, each encryption is applied in turn to obtain the ciphertext.

As discussed in earlier part of this document, the third method does not meet our requirements of MTAIBC.

In the remaining part of this document, we will focus on the second method.

The main technical block for implementing IBE and IBS algorithms from pairings on elliptic curves, which involve a single TA and a single ID, is the following pairing equation:

$$\mathbb{p}(R, Q) = \mathbb{p}(P, S) \quad (1)$$
$$\text{i.e., } \mathbb{p}(sP, Q) = \mathbb{p}(P, sQ) \quad (2)$$

For the purposes of using multiple TAs and/or multiple IDs in a more efficient way than an onion form, this pairing equation can be reconstructed as follows by address P, Q, s, R and S as P_i, Q_j, s_i, R_i and S_{ij} in MTAIBC:

$$\mathbb{p}(R_i, Q_j) = \mathbb{p}(P_i, S_{ij}) \quad (3)$$
$$\text{i.e., } \mathbb{p}(s_iP_i, Q_j) = \mathbb{p}(P_i, s_iQ_j) \quad (4)$$

4.2 A List of Cases

We distinguish the following a number of different cases for different organisation of using P, Q, s, R and S.

Case 0: including one TA and one user ID (the ordinary IBC)

Case 1: including one TA and multiple user IDs

Case 2: including mu ltiple TAs and one user ID

Case 3: including multiple TAs and multiple user IDs, where one ID is corresponding to one TA

Case 4: including multiple TAs and multiple user IDs, where one ID is corresponding to multiple TAs

Case 5: including multiple TAs each uses different P and one user ID.

Case 6: including multiple TAs each uses different P and multiple user IDs, where one ID is corresponding to one TA.

Case 7: including multiple TAs each uses different P and multiple user IDs, where one ID is corresponding to multiple TAs

The following table shows these seven cases briefly.

	P_i	s_i	R_i	Q_j	S_{ij}	$\mathbb{p}(s_iP_i, Q_j) = \mathbb{p}(P_i, s_iQ_j)$	$\mathbb{p}(R, Q) = \mathbb{p}(P, S)$
Case 0	P	s	sP	Q	sQ	$\mathbb{p}(sP, Q) = \mathbb{p}(P, sQ)$	$\mathbb{p}(sP, Q) = \mathbb{p}(P, sQ)$
Case 1	P	s	sP	Q_i	sQ_i	$\mathbb{p}(sP, Q_i) = \mathbb{p}(P, sQ_i)$	$\mathbb{p}(sP, \sum_{1 \le i \le n} b_iQ_i) = \mathbb{p}(P, \sum_{1 \le i \le n} b_isQ_i)$
Case 2	P	s_i	s_iP	Q	s_iQ	$\mathbb{p}(s_iP, Q) = \mathbb{p}(P, s_iQ)$	$\mathbb{p}(\sum_{1 \le i \le n} b_is_iP, Q) = \mathbb{p}(P, \sum_{1 \le i \le n} b_is_iQ)$
Case 3	P	s_i	s_iP	Q_i	s_iQ_i	$\mathbb{p}(s_iP, Q_i) = \mathbb{p}(P, s_iQ_i)$	$\Pi_{1 \le i \le n} \mathbb{p}(s_iP, Q_i)^{b_i} = \mathbb{p}(P, \sum_{1 \le i \le n} b_is_iQ_i)$
Case 4	P	s_i	s_iP	Q_j	s_iQ_j	$\mathbb{p}(s_iP, Q_j) = \mathbb{p}(P, s_iQ_j)$	$\Pi_{1 \le i, j \le n} \mathbb{p}(s_iP, Q_i)^{b_{ij}} = \mathbb{p}(P, \sum_{1 \le i, j \le n} b_{ii} S_{ij})$
Case 5	P_i	s_i	s_iP_i	Q	s_iQ	$\mathbb{p}(s_iP_i, Q) = \mathbb{p}(P_i, s_iQ)$	$\mathbb{p}(\sum_{1 \le i \le n} b_is_iP_i, Q) = \Pi_{1 \le i \le n} \mathbb{p}(P_i, s_iQ)^{b_i}$
Case 6	P_i	s_i	s_iP_i	Q_i	s_iQ_i	$\mathbb{p}(s_iP_i, Q_i) = \mathbb{p}(P_i, s_iQ_i)$	$\Pi_{1 \le i \le n} \mathbb{p}(s_iP_i, Q_i)^{b_i} = \Pi_{1 \le i \le n} \mathbb{p}(P_i, s_iQ_i)^{b_i}$
Case 7	P_i	s_i	s_iP_i	Q_j	s_iQ_j	$\mathbb{p}(s_iP_i, Q_j) = \mathbb{p}(P_i, s_iQ_j)$	$\Pi_{1 \le i, j \le n} \mathbb{p}(s_iP_i, Q_i)^{b_{ij}} = \Pi_{1 \le i, j \le n} \mathbb{p}(P_i, S_{ij})^{b_{ij}}$

In the following subclauses, we will give detailed discussion on Cases 1 - 7.

4.3 Case 1 ($P, R, Q_i, S_i = sQ_i$)

In this case, assume we have one fixed trust authority with its standard public/private key pair P and $R = sP$; and also assume we have a set of identifiers ID_i ($i = 1, \ldots, n$), each with private keys given by

$$S_i = sQ_i \text{ and } Q_i = H_1(ID_i).$$

Given an n bit string $b = (b_1, \ldots, b_n)$ we can then form the "virtual" identifier

$$Q = \sum_{1 \le i \le n} b_i Q_i,$$

the corresponding "virtual" private key

$$S = \sum_{1 \le i \le n} b_i S_i;$$

And therefore Equation 1, $\mathbb{p}(R, Q) = \mathbb{p}(P, S)$, becomes

$$\mathbb{p}(R, \sum_{1 \le i \le n} b_i Q_i) = \mathbb{p}(P, \sum_{1 \le i \le n} b_i S_i).$$

Example of applications

For some purposes, e.g., in order to encourage customers to access a service, the server makes an offer if any customer accesses the service three times, he will get a gift. The gift is encrypted under three IBC public keys, e.g., which are three random numbers, and are given to the customer in advance. Every time when a customer accesses the service, the customer gives the server one of the three random numbers. The server generates a relative IBC private key for him. After three times of access, the customer gets enough keys for decrypting his gift. In this applications,

$$Q_i = H_1(\text{the } i^{\text{th}} \text{ random number}) \ (i = 1, 2, 3),$$

and

$$\mathbb{p}(R, \sum_{1 \le i \le 3} Q_i) = \mathbb{p}(P, \sum_{1 \le i \le 3} sQ_i).$$

More detailed information of an IBE scheme for this example can be found in Annex A.1.

4.4 Case 2 ($P, R_i = s_iP, Q, S_i = s_iQ$)

In this case, there are n TAs each with their own standard public/private key pair P and $R_i = s_iP$. Suppose we have a fixed identifier ID and we obtain the n private keys corresponding to this identifier from the relevant trust authorities, i.e. we have

$$S_i = s_iQ, \text{ where } Q = H_1(ID).$$

Given an n bit string $b = (b_1, \ldots, b_n)$ we can then form the "virtual" trust authority with public key

$$R = \sum_{1 \le i \le n} b_i R_i,$$

the corresponding "virtual" private keys

$$s = \sum_{1 \le i \le n} b_i s_i \text{ and } S = \sum_{1 \le i \le n} b_i S_i;$$

And therefore Equation 1, $\mathbb{p}(R, Q) = \mathbb{p}(P, S)$, becomes

$$\mathbb{p}(\sum_{1 \le i \le n} b_i R_i, Q) = \mathbb{p}(P, \sum_{1 \le i \le n} b_i S_i), \text{ i.e.,}$$
$$\mathbb{p}(\sum_{1 \le i \le n} b_i s_i P, Q) = \mathbb{p}(P, \sum_{1 \le i \le n} b_i s_i Q).$$

Example of applications

In the United Kingdom every car needs to display a tax disk. This is purchased each year for a nominal fee, and essentially proves that at a given point in the year the owner of the car had car insurance and a certificate of roadworthiness for the car. We describe a possible online car tax disk dispenser.

We note the three obvious trust authorities:

- The ownership of the car is recorded by the Driver and Vehicle Licensing Agency (DVLA).
- The insurance certificate is produced by an insurance company, say AXA.
- The certificate of roadworthiness is produced by an accredited garage, say Joe's Garage.

The three trust authority public keys we denote by

$$R_1 = s_{\text{DVLA}} P, \; R_2 = s_{\text{AXA}} P, \text{ and } R_3 = s_{\text{Joes}} P.$$

Suppose the owner of the car with registration number X 675 AHO wished to obtain a new tax disk from the government. They could then log into some web site and claim that they owned the car, that they had insured it through AXA and that Joe's Garage had issued them with a certificate of roadworthiness. The government could then email the user an encrypted version of the tax disk, upon payment of some fee, where the encryption is under the virtual trust authority

$$R_1 + R_2 + R_3,$$

and the identifier is

$$Q = H_1(\text{X 675 AHO}).$$

The owner would need to obtain from each trust authority the corresponding private key (clearly date/year information needs to be added but we ignore that issue here for simplicity),

$$S_1 = s_{\text{DVLA}} Q, \; S_2 = s_{\text{AXA}} Q, S_3 = s_{\text{Joes}} Q.$$

The owner now adds these private keys together to form another private key

$$S = \sum_{1 \le i \le 3} S_i$$

with which they can decrypt the electronic form of the tax disk and print it on their printer.

$$\mathbb{p}(\sum_{1 \le i \le 3} R_i, Q) = \mathbb{p}(P, \sum_{1 \le i \le 3} S_i).$$

More detailed information of an IBE scheme for this example can be found in Annex A.2.

4.5 Case 3 (P, $R_i = s_i P$, Q_i, $S_i = s_i Q_i$)

In this case, there are n TAs each with their own standard public/private key pair $R_i = s_i P$; and there are n identifier IDs. Suppose we obtain the n private keys each corresponding to one of the n identifier IDs from the relevant trust authorities, i.e. we have

$$S_i = s_i Q_i, \text{ where } Q_i = H_1(\text{ID}_i).$$

Given an n bit string $b = (b_1, \ldots, b_n)$ we can then form the "virtual" trust authority with public key

$$R = \sum_{1 \le i \le n} b_i R_i ,$$

the corresponding "virtual" private key

$$s = \sum_{1 \le i \le n} b_i s_i, \text{ and } S = \sum_{1 \le i \le n} b_i S_i;$$

And therefore Equation 1, $\mathbb{p}(R, Q) = \mathbb{p}(P, S)$, becomes

$$\Pi_{1 \le i \le n} \mathbb{p}(R_i, Q_i)^{b_i} = \mathbb{p}(P, \sum_{1 \le i \le n} b_i S_i), \text{ i.e.,}$$
$$\Pi_{1 \le i \le n} \mathbb{p}(s_i P, Q_i)^{b_i} = \mathbb{p}(P, \sum_{1 \le i \le n} b_i s_i Q_i).$$

So for decryption we still add private keys together, but for encryption we need to multiply the ephemeral keys together, after they have been passed through the pairing.

Example of applications

Every TA knows a limited piece of information, and provides a limited part of the key generation service.

Alice wants to send Bob an encrypted disc that allows Bob to print his own property-transferred certificate. However she wants to make sure that Bob is known to a building society, is known to a bank/has the money, and is employee of HP. She chooses Nationwide, NWM, and HP as trusted authorities (with Bob's agreement) and chooses "Bob's mortgage", "Bob's bank account" and " Bob's employee number" as Bob's public keys. She encrypts the certificate disc under these public keys. To print the disc, Bob has to go to each of the trusted authorities for a private key. In this application,

$Q_1 = H_1$(Bob's mortgage), $Q_2 = H_1$(Bob's bank account), and $Q_3 = H_1$(Bob's employee number),

$$R_1 = s_{\text{Nationwide}} P, \; R_2 = s_{\text{NWM}} P, \text{ and } R_3 = s_{\text{HP}} P,$$

$$S_1 = s_{\text{Nationwide}} Q_1, \; S_2 = s_{\text{NWM}} Q_2, \text{ and } S_3 = s_{\text{HP}} Q_3,$$

and

$$\Pi_{1 \le i \le 3} \mathbb{p}(R_i, Q_i) = \mathbb{p}(P, \sum_{1 \le i \le 3} S_i).$$

More detailed information of an IBE scheme for this example can be found in Annex A.3.

4.6 Case 4 ($P, R_i = s_i P, Q_j, S_{ij} = s_i Q_j$)

In this case, assume that we have a set of TAs and a set of IDs, and then we have a set of atomic pairs (TA_i, ID_j, $i, j = 1, \ldots, n$). Each TA has their own standard public/private key pair $R_i = s_i P$. Suppose we obtain the n private keys each corresponding to one of the n identifier IDs from all of the trust authorities, i.e. we have

$$S_{ij} = s_i Q_j \text{ where } Q_j = H_1(\text{ID}_j).$$

Given an n bit string $b = (b_{11}, \ldots, b_{ij}, \ldots, b_m)$ we can then form the "virtual" trust authority with public key

$$R = \sum_{1 \le i \le n} b_{ij} R_i ,$$

the corresponding "virtual" private key

$$s = \sum_{1 \le i \le n} b_i s_i, \text{ and } S = \sum_{1 \le i, j \le n} b_{ij} S_{ii};$$

And therefore Equation 1, $\mathbb{p}(R, Q) = \mathbb{p}(P, S)$, becomes

$$\Pi_{1 \le i, j \le n} \mathbb{p}(R_i, Q_i)^{b_{ij}} = \mathbb{p}(P, \sum_{1 \le i, j \le n} b_{ij} S_{ij}), \text{ i.e.,}$$
$$\Pi_{1 \le i, j \le n} \mathbb{p}(s_i P, Q_i)^{b_{ij}} = \mathbb{p}(P, \sum_{1 \le i, j \le n} b_{ij} s_i Q_j).$$

Example of applications

Alice and Bob want to open a joint account in a community. They download an application form from the community's web side. Within the form, they are asked for information of their employment and address. They fill the form with the following information: Alice is HP employee; Bob is IBM employee and both of them are living in Bristol. The community sends them an encrypted document of the community membership. They have to work together to decrypt this document. The community chooses Alice-Bristol and Bob-Bristol as their IDs respectively; and chooses HP, IBM and Bristol local authority as TAs. In this application,

$$Q_1 = H_1(\text{Alice-Bristol}), \text{ and } Q_2 = H_1(\text{Bob-Bristol}),$$

$$R_1 = s_{\text{HP}} P, R_2 = s_{\text{IBM}} P, \text{ and } R_3 = s_{\text{Bristol}} P,$$

$$S_{11} = s_{\text{HP}} Q_1, S_{22} = s_{\text{IBM}} Q_2, S_{31} = s_{\text{Bristol}} Q_1, \text{ and } S_{32} = s_{\text{Bristol}} Q_2,$$

$$b_{11}, b_{22}, b_{31}, b_{32} = 1, b_{12}, b_{21} = 0,$$

and

$$\Pi_{1 \le i \le 3, 1 \le j \le 2} \mathbb{p}(R_i, Q_j)^{b_{ij}} = \mathbb{p}(P, \sum_{1 \le i \le 3, 1 \le j \le 2} b_{ij} S_{ij}).$$

More detailed information of an IBE scheme for this example can be found in Annex A.4.

4.7 Case 5 (P_i, $R_i = s_i P_i$, Q, $S_i = s_i Q$)

In this case, there are n TAs each with their own point P_i and standard public/private key pair $R_i = s_i P_i$; and there is one fixed identifier ID. Suppose we obtain the n private keys to the ID each from the relevant trust authorities, i.e. we have

$$S_i = s_i Q \text{ where } Q = H_1(\text{ID}).$$

Given an n bit string $b = (b_1, \ldots, b_n)$ we can then form the "virtual" trust authority with public key

$$R = \sum_{1 \le i \le n} b_i R_i,$$

the corresponding "real" private key

$$S = \sum_{1 \le i \le n} b_i S_i;$$

And therefore Equation 1, $\mathbb{p}(R, Q) = \mathbb{p}(P, S)$, becomes

$$\mathbb{p}(\sum_{1 \le i \le n} b_i R_i, Q) = \Pi_{1 \le i \le n} \mathbb{p}(P_i, S_i)^{b_i}.$$
$$\mathbb{p}(\sum_{1 \le i \le n} b_i s_i P_i, Q) = \Pi_{1 \le i \le n} \mathbb{p}(P_i, s_i Q)^{b_i}.$$

So for decryption we still add private keys together, but for encryption we need to multiply ephemeral keys together, after they have been passed through the pairing.

Example of applications

The same example of Case 2.

More detailed information of an IBE scheme for this example can be found in Annex A.5.

4.8 Case 6 (P_i, $R_i = s_iP_i$, Q_i, $S_i = s_iQ_i$)

In this case, there are n TAs each with their own point P_i and standard public/private key pair $R_i = s_iP_i$; and there is n identifier IDs. Suppose we obtain the n private keys each corresponding to one of the n identifier IDs from the relevant trust authorities, i.e. we have

$$S_i = s_iQ_i \text{ where } Q_i = H_1(\mathrm{ID}_i).$$

Given an n bit string $b = (b_1, \ldots, b_n)$ we can then form the "virtual" trust authority with public key

$$R = \sum_{1 \le i \le n} b_iR_i,$$

the corresponding "virtual" private key

$$S = \sum_{1 \le i \le n} b_iS_i;$$

And therefore Equation 1, $\mathbb{p}(R, Q) = \mathbb{p}(P, S)$, becomes

$$\Pi_{1 \le i \le n} \mathbb{p}(R_i, Q_i)^b = \Pi_{1 \le i \le n} \mathbb{p}(P_i, S_i)^{b_i}.$$
$$\Pi_{1 \le i \le n} \mathbb{p}(s_iP_i, Q_i)^b = \Pi_{1 \le i \le n} \mathbb{p}(P_i, s_iQ_i)^{b_i}.$$

So for decryption we still add private keys together, but for encryption we need to multiply ephemeral keys together, after they have been passed through the pairing.

Example of applications

The same example of Case 3.

More detailed information of an IBE scheme for this example can be found in Annex A.6.

4.9 Case 7 (P_i, $R_i = s_iP_i$, Q_j, $S_{ij} = s_iQ_j$)

In this case, there are n TAs each with their own point P_i and standard public/private key pair $R_i = s_iP_i$; and there is n identifier IDs. Suppose we obtain the n private keys each corresponding to one of the n identifier IDs from each of the n TAs, i.e. we have

$$S_{ij} = s_iQ_j \text{ where } Q_j = H_1(\mathrm{ID}_j).$$

Given an n bit string $b = (b_1, \ldots, b_n)$ we can then form the "virtual" trust authority with public key

$$R = \sum_{1 \le i \le n} b_iR_i,$$

the corresponding "real" private key

$$S = \sum_{1 \le i \le n} b_i S_i;$$

And therefore Equation 1, $\mathbb{p}(R, Q) = \mathbb{p}(P, S)$, becomes

$$\Pi_{1 \le i \le n,\ 1 \le j \le n}\ \mathbb{p}(R_i, Q_i)^{b_i} = \Pi_{1 \le i \le n,\ 1 \le j \le n}\ \mathbb{p}(P_i, S_i)^{b_i}.$$
$$\Pi_{1 \le i \le n,\ 1 \le j \le n}\ \mathbb{p}(s_i P_i, Q_i)^{b_i} = \Pi_{1 \le i \le n,\ 1 \le j \le n}\ \mathbb{p}(P_i, s_i Q_j)^{b_i}.$$

So for decryption we still add private keys together, but for encryption we need to multiply ephemeral keys together, after they have been passed through the pairing.

Example of applications

The same example of Case 4.

More detailed information of an IBE scheme for this example can be found in Annex A.7.

5 Summary

In this document, we have discussed the issue of multiple trusted authorities in identifier based cryptography (MTAIBC); and we have considered seven different cases in one level TAs with operation of addition.

Annex A

Examples of the IBE Scheme with MTA

In this annex, we give seven examples of MTAIBC schemes, each for one case described in Section 4. All of these examples are based on the Boneh and Franklin's IBE scheme [BF01a].

A.1 An Example Scheme of Case 1

Summary: Alice encrypts a gift $m \in \{0,1\}^n$ for Bob, which Bob can decrypt if he has 3 private keys S_i (i = 1, 2, 3), each respectively issued by a TA (which is Alice herself in this example) corresponding to a public key Q_i (i = 1, 2, 3).

Setup TA. TA should do the following:

1. Choose a large (at least 512-bits) prime p such that $p = 2 \bmod 3$ and $p = 6q - 1$ for some prime $q > 3$. Let E be the elliptic curve defined by $y^2 = x^3 + 1$ over $\mathbb{F}_p$.
2. Choose an arbitrary $P \in E/\mathbb{F}_p$ of order q.
3. Pick four hash functions: H_1: $\{0,1\}^* \rightarrow \mathbb{F}_p$; H_2: $\mathbb{F}_{p^2} \rightarrow \{0,1\}^n$ for some n; H_3: $\{0,1\}^n \times \{0,1\}^n \rightarrow \mathbb{Z}^*_q$, and H_4: $\{0,1\}^n \rightarrow \{0,1\}^n$.
4. Selects a random $s \in \mathbb{Z}^*_q$ and set $R = sP$.

Register Bob. TA should do the following:

1. Choose three random numbers $\in \{0,1\}^n$
2. Compute MapToPoint $Q_i = H_1$(the i^{th} random number) (i = 1, 2, 3) $\in E/\mathbb{F}_p$ of order q.
3. Set the private key $S_i = sQ_i$.

Encryption. Alice should do the following:

1. Select a random $s \in \{0,1\}^n$.
2. Compute $r = H_3(s, m)$.
3. Compute $U = rP$.
4. Compute $g_{\text{ID}} = \mathbb{p}(\sum_{1 \le i \le 3} Q_i, R) \in \mathbb{F}_{p^2}$ – may be precomputed.
5. Compute $V = s \oplus H_2(g_{\text{ID}}{}^r)$.
6. Compute $W = m \oplus H_4(s)$.
7. Set the ciphertext to be $C = (U, V, W)$.

Decryption. To cover m from C, Bob should do the following:

1. Test $U \in E/\mathbb{F}_p$ of order q.
2. Compute $x = \mathbb{p}(U, \sum_{1 \le i \le 3} S_i)$
3. Compute $s = V \oplus H_2(x)$.
4. Compute $m = W \oplus H_4(s)$.
5. Compute $r = H_3(s, m)$.
6. Check $U = rP$.

A.2 An Example Scheme of Case 2

Summary: Alice encrypts a car tax message $m \in \{0,1\}^n$ for Bob, which Bob can decrypt if he has 3 private keys S_i (i = 1, 2, 3), each respectively issued by a TA_i (i = 1, 2, 3) corresponding to the same public key Q.

Setup TA_i. TA_i should do the following:

1. Accept a large (at least 512-bits) prime p such that $p = 2 \bmod 3$ and $p = 6q - 1$ for some prime $q > 3$. Let E be the elliptic curve defined by $y^2 = x^3 + 1$ over $\mathbb{F}_p$.
2. Accept an arbitrary $P \in E/\mathbb{F}_p$ of order q.

3. Accept four hash functions: H_1: {0,1}* $\rightarrow \mathbb{F}_p$; H_2: $\mathbb{F}_{p^2} \rightarrow$ {0,1}^{n} for some n; H_3: {0,1}$^n \times$ {0,1}^{n} $\rightarrow \mathbb{Z}^*_q$, and H_4: {0,1}$^n \rightarrow$ {0,1}^{n}.
4. Selects a random $s_i \in \mathbb{Z}^*_q$ and set $R_i = s_iP$.

Register Bob. TA$_i$ should do the following:

1. Compute MapToPoint $Q = H_1$(X 675 AHO) $\in E/\mathbb{F}_p$ of order q.
2. Set the private key $S_i = s_iQ$.

Encryption. Alice should do the following:

1. Compute MapToPoint $Q = H_1$(X 675 AHO) $\in E/\mathbb{F}_p$ of order q.
2. Select a random $s \in$ {0,1}^{n}.
3. Compute $r = H_3(s, m)$.
4. Compute $U = rP$.
5. Compute $g_{ID} = \mathbb{p}(Q, \sum_{1 \le i \le 3} R_i) \in \mathbb{F}_{p^2}$ – may be precomputed.
6. Compute $V = s \oplus H_2(g_{ID}{}^r)$.
7. Compute $W = m \oplus H_4(s)$.
8. Set the ciphertext to be $C = (U, V, W)$.

Decryption. To cover m from C, Bob should do the following:

1. Test $U \in E/\mathbb{F}_p$ of order q.
2. Compute $x = \mathbb{p}(U, \sum_{1 \le i \le 3} S_i)$
3. Compute $s = V \oplus H_2(x)$.
4. Compute $m = W \oplus H_4(s)$.
5. Compute $r = H_3(s, m)$.
6. Check $U = rP$.

A.3 An Example Scheme of Case 3

Summary: Alice encrypts a property-transferred certificate $m \in$ {0,1}^{n} for Bob, which Bob can decrypt if he has 3 private keys S_i (i = 1, 2, 3), each respectively issued by a TA$_i$ (i = 1, 2, 3) corresponding to a public key Q_i (i = 1, 2, 3), based on ID$_1$ = Bob's mortgage; ID$_2$ = Bob's bank account; and ID$_3$ = Bob's employee number.

Setup TA$_i$. TA$_i$ should do the following:

1. Accept a large (at least 512-bits) prime p such that $p = 2 \bmod 3$ and $p = 6q - 1$ for some prime $q > 3$. Let E be the elliptic curve defined by $y^2 = x^3 + 1$ over $\mathbb{F}_p$.
2. Accept an arbitrary $P \in E/\mathbb{F}_p$ of order q.
3. Accept four hash functions: H_1: {0,1}* $\rightarrow \mathbb{F}_p$; H_2: $\mathbb{F}_{p^2} \rightarrow$ {0,1}^{n} for some n; H_3: {0,1}$^n \times$ {0,1}^{n} $\rightarrow \mathbb{Z}^*_q$, and H_4: {0,1}$^n \rightarrow$ {0,1}^{n}.
4. Selects a random $s_i \in \mathbb{Z}^*_q$ and set $R_i = s_iP$.

Register Bob. TA$_i$ should do the following:

1. Compute MapToPoint $Q_i = H_1$(ID$_i$) $\in E/\mathbb{F}_p$ of order q.
2. Set the private key $S_i = s_iQ_i$.

Encryption. Alice should do the following:

1. Compute MapToPoint $Q_i = H_1$(ID$_i$) $\in E/\mathbb{F}_p$ of order q.
2. Select a random $s \in$ {0,1}^{n}.
3. Compute $r = H_3(s, m)$.
4. Compute $U = rP$.
5. Compute $g_{ID} = \Pi_{1 \le i \le 3} \mathbb{p}(R_i, Q_i) \in \mathbb{F}_{p^2}$ – may be precomputed.
6. Compute $V = s \oplus H_2(g_{ID})$.
7. Compute $W = m \oplus H_4(s)$.
8. Set the ciphertext to be $C = (U, V, W)$.

Decryption. To cover m from C, Bob should do the following:

1. Test $U \in E/\mathbb{F}_p$ of order q.
2. Compute $x = \mathbb{p}(U, \sum_{1 \le i \le 3} S_i)$
3. Compute $s = V \oplus H_2(x)$.
4. Compute $m = W \oplus H_4(s)$.
5. Compute $r = H_3(s, m)$.
6. Check $U = rP$.

A.4 An Example Scheme of Case 4

Summary: A communitiy (which Alice and Bob want to join in) encrypts a community membership certificate $m \in \{0,1\}^n$ for Alice and Bob, which they can decrypt if they has 4 private keys S_{ij} (ij = 11, 22, 31, 32), each respectively issued by a TA_i (i = 1, 2, 3) corresponding to a public key Q_j (j = 1, 2), based on ID_1 = Alice-Bristol; and ID_2 = Bob - Bristol.

Setup TA_i. TA_i should do the following:

1. Accept a large (at least 512-bits) prime p such that $p = 2 \bmod 3$ and $p = 6q - 1$ for some prime $q > 3$. Let E be the elliptic curve defined by $y^2 = x^3 + 1$ over $\mathbb{F}_p$.
2. Accept an arbitrary $P \in E/\mathbb{F}_p$ of order q.
3. Accept four hash functions: $H_1: \{0,1\}^* \to \mathbb{F}_p$; $H_2: \mathbb{F}_{p^2} \to \{0,1\}^n$ for some n; $H_3: \{0,1\}^n \times \{0,1\}^n \to \mathbb{Z}^*_q$, and $H_4: \{0,1\}^n \to \{0,1\}^n$.
4. Selects a random $s_i \in \mathbb{Z}^*_q$ and set $R_i = s_i P$.

Register Bob. TA_i should do the following:

1. Compute MapToPoint $Q_j = H_1(ID_j) \in E/\mathbb{F}_p$ of order q.
2. Set the private key $S_{ij} = s_i Q_j$.

Encryption. The community should do the following:

1. Compute MapToPo int $Q_j = H_1(ID_j) \in E/\mathbb{F}_p$ of order q.
2. Select a random $s \in \{0,1\}^n$.
3. Compute $r = H_3(s, m)$.
4. Compute $U = rP$.
5. Compute $g_{ID} = \Pi_{1 \le i \le 3, 1 \le j \le 2} \mathbb{p}(R_i, Q_j)^{b_{ij}}$ ($b_{11}, b_{22}, b_{31}, b_{32} = 1, b_{12}, b_{21} = 0$) $\in \mathbb{F}_{p^2}$ – may be precomputed.
6. Compute $V = s \oplus H_2(g_{ID})$.
7. Compute $W = m \oplus H_4(s)$.
8. Set the ciphertext to be $C = (U, V, W)$.

Decryption. To cover m from C, Alice and Bob should do the following:

1. Test $U \in E/\mathbb{F}_p$ of order q.
2. Compute $x = \mathbb{p}(U, \sum_{1 \le i \le 3, 1 \le j \le 2} b_{ij} S_i)$ ($b_{11}, b_{22}, b_{31}, b_{32} = 1, b_{12}, b_{21} = 0$)
3. Compute $s = V \oplus H_2(x)$.
4. Compute $m = W \oplus H_4(s)$.
5. Compute $r = H_3(s, m)$.
6. Check $U = rP$.

A.5 An Example Scheme of Case 5

Summary: Alice encrypts a car tax message $m \in \{0,1\}^n$ for Bob, which Bob can decrypt if he has 3 private keys S_i (i = 1, 2, 3), each respectively issued by a TA_i (i = 1, 2, 3) corresponding to the same public key Q.

Setup TA_i. TA_i should do the following:

1. Accept a large (at least 512-bits) prime p such that $p = 2 \bmod 3$ and $p = 6q - 1$ for some prime $q > 3$. Let E be the elliptic curve defined by $y^2 = x^3 + 1$ over $\mathbb{F}_p$.
2. Choose an arbitrary $P_i \in E/\mathbb{F}_p$ of order q.
3. Accept four hash functions: H_1: $\{0,1\}^* \to \mathbb{F}_p$; H_2: $\mathbb{F}_{p^2} \to \{0,1\}^n$ for some n; H_3: $\{0,1\}^n \times \{0,1\}^n \to \mathbb{Z}^*_q$, and H_4: $\{0,1\}^n \to \{0,1\}^n$.
4. Selects a random $s_i \in \mathbb{Z}^*_q$ and set $R_i = s_i P_i$.

Register Bob. TA_i should do the following:

1. Compute MapToPoint $Q = H_1$(X 675 AHO) $\in E/\mathbb{F}_p$ of order q.
2. Set the private key $S_i = s_i Q$.

Encryption. Alice should do the following:

1. Compute MapToPoint $Q = H_1$(X 675 AHO) $\in E/\mathbb{F}_p$ of order q.
2. Select a random $s \in \{0,1\}^n$.
3. Compute $r = H_3(s, m)$.
4. Compute $U_i = rP_i$.
5. Compute $g_{ID} = \mathbb{p}(Q, \sum_{1 \le i \le 3} R_i) \in \mathbb{F}_{p^2}$ – may be precomputed.
6. Compute $V = s \oplus H_2(g_{ID^r})$.
7. Compute $W = m \oplus H_4(s)$.
8. Set the ciphertext to be $C = (U_1, U_2, U_3, V, W)$.

Decryption. To cover m from C, Bob should do the following:

1. Test $U_i \in E/\mathbb{F}_p$ of order q.
2. Compute $x = \prod_{1 \le i \le 3} \mathbb{p}(U_i, S_i)$
3. Compute $s = V \oplus H_2(x)$.
4. Compute $m = W \oplus H_4(s)$.
5. Compute $r = H_3(s, m)$.
6. Check $U_i = rP_i$.

A.6 An Example Scheme of Case 6

Summary: Alice encrypts a property-transferred certificate $m \in \{0,1\}^n$ for Bob, which Bob can decrypt if he has 3 private keys S_i (i = 1, 2, 3), each respectively issued by a TA_i (i = 1, 2, 3) corresponding to a public key Q_i (i = 1, 2, 3), based on ID_1 = Bob's mortgage; ID_2 = Bob's bank account; and ID_3 = Bob's employee number.

Setup TA_i. TA_i should do the following:

1. Accept a large (at least 512-bits) prime p such that $p = 2 \bmod 3$ and $p = 6q - 1$ for some prime $q > 3$. Let E be the elliptic curve defined by $y^2 = x^3 + 1$ over $\mathbb{F}_p$.
2. Choose an arbitrary $P_i \in E/\mathbb{F}_p$ of order q.
3. Accept four hash functions: H_1: $\{0,1\}^* \to \mathbb{F}_p$; H_2: $\mathbb{F}_{p^2} \to \{0,1\}^n$ for some n; H_3: $\{0,1\}^n \times \{0,1\}^n \to \mathbb{Z}^*_q$, and H_4: $\{0,1\}^n \to \{0,1\}^n$.
4. Selects a random $s_i \in \mathbb{Z}^*_q$ and set $R_i = s_i P_i$.

Register Bob. TA_i should do the following:

1. Compute MapToPoint $Q_i = H_1(ID_i) \in E/\mathbb{F}_p$ of order q.
2. Set the private key $S_i = s_i Q_i$.

Encryption. Alice should do the following:

1. Compute MapToPoint $Q_i = H_1(ID_i) \in E/\mathbb{F}_p$ of order q.
2. Select a random $s \in \{0,1\}^n$.
3. Compute $r = H_3(s, m)$.
4. Compute $U_i = rP_i$.
5. Compute $g_{ID} = \prod_{1 \le i \le 3} \mathbb{p}(R_i, Q_i) \in \mathbb{F}_{p^2}$ – may be precomputed.

6. Compute $V = s \oplus H_2(g_{ID'})$.
7. Compute $W = m \oplus H_4(s)$.
8. Set the ciphertext to be $C = (U_1, U_2, U_3, V, W)$.

Decryption. To cover m from C, Bob should do the following:

1. Test $U_i \in E/\mathbb{F}_p$ of order q.
2. Compute $x = \prod_{1 \le i \le 3} \mathbb{p}(U_i, S_i)$
3. Compute $s = V \oplus H_2(x)$.
4. Compute $m = W \oplus H_4(s)$.
5. Compute $r = H_3(s, m)$.
6. Check $U_i = rP_i$.

A.7 An Example Scheme of Case 7

Summary: A community (which Alice and Bob want to join in) encrypts a community membership certificate $m \in \{0,1\}^n$ for Alice and Bob, which they can decrypt if they has 4 private keys S_{ij} (ij = 11, 22, 31, 32), each respectively issued by a TA_i (i = 1, 2, 3) corresponding to a public key Q_j (j = 1, 2), based on ID_1 = Alice-Bristol; and ID_2 = Bob - Bristol.

Setup TA_i. TA_i should do the following:

1. Accept a large (at least 512-bits) prime p such that $p = 2 \bmod 3$ and $p = 6q - 1$ for some prime $q > 3$. Let E be the elliptic curve defined by $y^2 = x^3 + 1$ over $\mathbb{F}_p$.
2. Choose an arbitrary $P_i \in E/\mathbb{F}_p$ of order q.
3. Accept four hash functions: $H_1: \{0,1\}^* \to \mathbb{F}_p$; $H_2: \mathbb{F}_{p^2} \to \{0,1\}^n$ for some n; $H_3: \{0,1\}^n \times \{0,1\}^n \to \mathbb{Z}^*_q$, and $H_4: \{0,1\}^n \to \{0,1\}^n$.
4. Selects a random $s_i \in \mathbb{Z}^*_q$ and set $R_i = s_i P_i$.

Register Bob. TA_i should do the following:

1. Compute MapToPoint $Q_j = H_1(ID_j) \in E/\mathbb{F}_p$ of order q.
2. Set the private key $S_{ij} = s_i Q_j$.

Encryption. The community should do the following:

1. Compute MapToPoint $Q_j = H_1(ID_j) \in E/\mathbb{F}_p$ of order q.
2. Select a random $s \in \{0,1\}^n$.
3. Compute $r = H_3(s, m)$.
4. Compute $U_i = rP_i$.
5. Compute $g_{ID} = \prod_{1 \le i \le 3, 1 \le j \le 2} \mathbb{p}(R_i, Q_j)^{b_{ij}}$ ($b_{11}, b_{22}, b_{31}, b_{32} = 1$, $b_{12}, b_{21} = 0$) $\in \mathbb{F}_{p^2}$ – may be precomputed.
6. Compute $V = s \oplus H_2(g_{ID'})$.
7. Compute $W = m \oplus H_4(s)$.
8. Set the ciphertext to be $C = (U_1, U_2, U_3, V, W)$.

Decryption. To cover m from C, Alice and Bob should do the following:

1. Test $U_i \in E/\mathbb{F}_p$ of order q.
2. Compute $x = \prod_{1 \le i \le 3, 1 \le j \le 2} \mathbb{p}(U_i, S_{ij})^{b_{ij}}$ ($b_{11}, b_{22}, b_{31}, b_{32} = 1$, $b_{12}, b_{21} = 0$)
3. Compute $s = V \oplus H_2(x)$.
4. Compute $m = W \oplus H_4(s)$.
5. Compute $r = H_3(s, m)$.
6. Check $U_i = rP_i$.

Bibliography

[BF01a] D. Boneh and M. Franklin. Identity based encryption from the Weil pairing. In *Advances in Cryptology - CRYPTO 2001*, LNCS 2139, pp. 213-229, Springer-Verlag, 2001.

[BF01b] D.Boneh and M. Franklin. Efficient generation of shared RSA keys. In *Journal of the ACM (JACM)*, Vol. 48, pp. 702—722, July 2001.

[BLS01] D. Boneh, B. Lynn and H. Shacham. Short signatures from the Weil pairing. In *Advances in Cryptology - ASIACRYPT 2001*, LNCS 2248, pp. 514-532, Springer-Verlag, 2001.

[Bo88] C. Boyd. Some applications of multiple key ciphers. In *Advances in Cryptology - EUROCRYPT '88*, LNCS 330, pp. 455-467, Springer-Verlag, 1988.

[Bo89] C. Boyd. A new multiple key cipher and an improved voting scheme. In *Advances in Cryptology - EUROCRYPT '89*, LNCS 434, pp. 617-625, Springer-Verlag, 1989.

[CC02] J. C. Cha and J. H. Cheon. An identity-based signature from gap Diffie-Hellman groups. In *Proceedings of the 6th Annual International Workshop on Practice and Theory in Public Key Cryptography*, LNCS 2567, pp. 18-30, Springer-Verlag, 2002.

[CGMW97] L. Chen, D. Gollmann, C. Mitchell and P. Wild. Secret sharing with reusable polynomials. In *Proceeding of the Second Australasian Conference on Information Security and Privacy*, LNCS 1270, pp. 183-193, Springer-Verlag, 1997.

[CHSS02] L. Chen, K. Harrison, N.P. Smart and D. Soldera. Applications of multiple trust authorities in pairing based cryptosystems. In *Proceedings of Infrastructure Security Conference 2002*, LNCS 2437, pp. 260-275, Springer-Verlag, 2002.

[CHMSS02] L. Chen, K. Harrison, A. Moss, N.P. Smart and D. Soldera. Certification of public keys within an identifier based system. In *Proceedings of Information Security Conference 2002*, LNCS 2433, pages 322-333, Springer-Verlag, 2002.

[CK02] L. Chen and C. Kudla. Identity based authenticated key agreement protocols from pairings. Preprint 2002.

[Co01] C. Cocks. An identity based encryption scheme based on quadratic residues. In *Proceedings of the 8th IMA International Conference on Cryptography and Coding*, LNCS 2260, pp. 360-363, Springer-Verlag, 2001.

[De87] Y. Desmedt. Society and group oriented cryptography: a new concept. In *Advances in Cryptology - CRYPTO '87*, LNCS 293, pp. 120-127, Springer-Verlag, 1987.

[DQ86] Y. Desmedt and J.-J. Quisquater. Public-key systems based on the difficulty of tampering (Is there a difference between DES and RSA?). In *Advances in Cryptology – CRYPTO '86*, LNCS 263, pp. 111-117, Springer-Verlag, 1987.

[FMR99] G. Frey, M. Müller, and H. Rück. The Tate pairing and the discrete logarithm applied to elliptic curve cryptosystems. *IEEE Transactions on Information Theory*, **45**(5):1717–1719, 1999.

[FO99] E. Fujisaki and T. Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *Advances in Cryptology - CRYPTO '99*, LNCS 1666, pp. 537-554, Springer-Verlag, 1999.

[Ga01] S. Galbraith. Supersingular curves in cryptography. In *Advances in Cryptology – Asiacrypt' 01*, LNCS 2248, pp. 495-513, Springer-Verlag, 2001.

[Ga02] S. Galbraith. Private communication, 2002.

[GP90] M. Girault and J.C. Paillès. An identity-based scheme providing zero-knowledge authentication and authenticated key exchange. In *Proceeedings of ESORICS '90*, pp. 173-184, 1990.

[GQ88] L.C. Guillou and J.-J. Quisquater. A practical zero-knowledge protocol fitted to security microprocessor minimizing both transmission and memory. In *Advances in Cryptology - EUROCRYPT '88*, LNCS 330, pp. 123-128, Springer-Verlag, 1988.

[GS02] C. Gentry and A. Silverberg. Hierarchical ID-Based Cryptography. In *Proceedings of Advances in Cryptology - Asiacrypt 2002*, LNCS 2501, pp. 548-566, Springer-Verlag, 2002.

[GS94] M. Girault and J. Stern. On the length of cryptographic hash-values used in identification schemes. In *Advances in Cryptology - CRYPTO '94*, LNCS 839, pp. 202-215, Springer-Verlag, 1994.

[He02] F. Hess. Efficient identity based signature schemes based on pairings. In *Proceedings of the Ninth Annual Workshop on Selected Areas in Cryptography*, 2002.

[HL02] J. Horwitz and B. Lynn. Towards hierarchical identity-based encryption. In *Advances in Cryptology - EUROCRYPT 2002*, LNCS 2332, pp. 466-481, Springer-Verlag, 2002.

[ISO/IEC 9594-8] ISO/IEC 9594-8:2001(the 4th edition), Information technology –Open Systems Interconnection – The Directory: Public-key and attribute certificate frameworks. International Organization for Standardization, Geneva, Switzerland, 2001.

[ISO/IEC 11770-3] ISO/IEC 11770-3. Information technology – Security Techniques – Key management – Part 3: Mechanisms using asymmetric techniques. International Organization for Standardization, Geneva Switzerland, 1999.

[ISO/IEC 14888-2] ISO/IEC 14888-2: 1998, Information technology – Security techniques – Digital signatures with appendix – Part 2: Identity – based mechanisms. International Organization for Standardization, Geneva, Switzerland, 1998.

[MCH03] M. Casassa Mont, L. Chen and K. Harrison. Turning the tables: making cryptography work for you. To appear in *Proceedings of HP Techonology Conference 2003*.

[MOV93] A.J. Menezes, T. Okamoto and S. Vanstone. Reducing elliptic curve logarithms to logarithms in a finite field. *IEEE Transactions on Information Theory*, **39**:1639-1646, 1993.

[MY91] U.M. Maurer and Y. Yacobi. Non-interactive public key cryptography. In *Advances in Cryptology – EUROCRYPT '91*, LNCS 547, pp. 498 – 507, Spring-Verlag, 1991.

[Ok86] E. Okamoto. Proposal for identity-based key distribution system. *Electronics Letters*, **22**:1283-1284, 1986.

[Ok87] E. Okamoto. Key distribution system based on identification information. In *Advances in Cryptology - CRYPTO '87*, LNCS 293, pp. 194-202, Spring-Verlag, 1987.

[Pa02] K. Paterson. ID-based signatures from pairings on elliptic curves. *Electronic Letters*, 38(18):1025-1026, 2002.

[Pi96] R.G.E. Pinch. On-line multiple secret sharing. Electronics Letters 32:1087-1088, 1996.

[Si94] J.H. Silverman. Advanced topics in the arithmetic of elliptic curves. GTM 151, ISBN 0-387-94325-0, Springer-Verlag, 1994.

[Sh79] A. Shamir. How to share a secret. *Communications of the ACM*, 22(11):612-613, 1979.

[Sh84] A. Shamir. Identity-based cryptosystems and signature schemes. In *Advances in Cryptology - CRYPTO '84*, LNCS 196, pp 47-53, Springer-Verlag, 1985.

[Sm02] N.P. Smart. An identity based authenticated key agreement protocol based on the Weil pairing. *Electronics Letters*, **38**:630-632, 2002.

[So02] D. Soldera. Private communication, 2002.

[SOK00] R. Sakai, K. Ohgishi and M. Kasahara. Cryptosystems based on pairing. In *Proceedings of the 2000 Symposium on Cryptography and Information Security*, Okinawa, Japan, January 26-28, 2000.

[Ta88] H. Tanaka. A realization scheme for the identity-based cryptosystem. In *Advances in Cryptology – CRYPTO '87*, LNCS 293, pp. 340-349, Springer-Verlag, 1988.

[TO91] K. Tanaka and E. Okamoto. Key distribution system for mail systems using ID-related information directory. *Computers & Security*, **10**:25-33, 1991.

[VZ97] S.A. Vanstone and R.J.Zuccherato. Elliptic curve cryptosystems using curves of smooth order over the ring $\mathbb{Z}_n$. *IEEE Transactions on Information Theory*, 43(4):1231-1237, July 1997.