Two Protocols from the Tate Pairing

Noel McCullagh and Michael Scott,

School of Computing,
Dublin City University,
Glasnevin, Dublin 9.
Ireland.
noel.mccullagh@computing.dcu.ie **
mike@computing.dcu.ie

Abstract. In this paper we present two new protocols from the Tate Pairing. The first is a secret handshake, in which we introduce a covert channel into Smarts "An identity based key agreement protocol based on the weil pairing" and the second is an efficient Signcryption scheme for low bandwidth channels.

keywords: handshake, key-agreement, secret, covert, identity based, signcryption.

1 Introduction

Handshakes are one of the most often used protocols in cryptography. Two of the fundamentals of secure communication are authentication - that you are sure who you are communicating with, and confidentiality - that the data can only be read by the intended recipient. The basic idea of a cryptographic handshake is that two unauthenticated parties enter into a protocol. The outcome of which should be that either one or both of the parties is authenticated, and that both parties have formulated a shared secret with which to encrypt future data; an example of this is the SSL protocol.

Using traditional PKI methods these protocols relied on the parties obtaining each others certificates, extracting each others public keys, checking certificate chains, which may involve many signature verifications, and finally generating a shared secret. Identity Based Encryption (IBE), an idea first proposed by Shamir in 1984[1], made possible first by Cocks [2], and then Boneh and Franklin [3], greatly simplifies this process. The persons online identity, i.e. their email address, becomes their certified public key. This greatly reduces the overhead when compared with PKI, both computationally and in terms of key management. This is currently an area of very active research, with many identity based security protocols being proposed [4,5,6,7].

** This author wishes to thank Enterprise Ireland for their support with this research under grant IF/2002/0312/N.

Secret Handshakes have been used since the middle ages to allow people that have never met to authenticate each other as members of a secret group. Balfanz et al have proposed a secret handshake scheme from bilinear maps [8]. This scheme uses a non-interactive key agreement protocol to establish a shared secret if both parties are members of a group. As it is non-interactive they do leak any information about their status as members of the secret group. One problem with this scheme is that if both parties are not members of the same group they do not establish a shared secret. In this situation communication between the parties terminates abruptly, and they may become suspicious of each other. We feel one of the properties of a physical secret handshake is missing, that the secret handshake itself should not be recognised as such by a non-member of the particular group. Therefore it should be as similar as possible to a normal handshake, whilst still being different enough to allow members of the secret group to recognise their handshake. The traditional indication that someone who has just received the handshake is a member of the group is that they reply with a secret codeword. Obviously non-members, not recognising the significance of the handshake and mistaking it for a regular handshake, will not know to do this and so will unknowingly give themselves away as non-members.

The idea of the proposed cryptographic secret handshake allows two people who have never met to calculate a shared secret. However, if one or neither of them is in a particular group then they will share a "standard secret". In contrast, if both of them are in the same secret group, then they will share a secret unique to them as members of that particular group.

2 Mathematical Preliminaries

An elliptic curve $\mathbb{E}(\mathbb{F}_{q^k})$ is the set of solutions (x, y) over the field $\mathbb{F}_{q^k}$ to an equation of the form $y^2 = x^3 + Ax + B$, together with an additional point at infinity, denoted O. There exists an abelian group law on $\mathbb{E}$. There are explicit formulas for computing the coordinates of a point $P_3 = P_1 + P_2$ from the coordinates of P_1 and P_2. Scalar multiplication of a point is defined as the repeated addition of a point to itself n times, e.g. $3P_1 = P_1 + P_1 + P_1$.

The number of points of an elliptic curve $\mathbb{E}(\mathbb{F}_{q^k})$ is called the order of the curve over the field $\mathbb{F}_{q^k}$. A point P has order r if $rP = O$. The set of all points of order r in $\mathbb{E}$ is denoted $\mathbb{E}[r]$. This is arranged as cyclic subgroups of prime order r. The order of a point always evenly divides the curve order. A subgroup $\mathbb{G}$ of an elliptic curve is said to have embedding degree *(a.k.a security multiplier)* k if its order r divides $q^k - 1$ for the smallest possible k. We assume $k > 1$. [9]

Our system uses the Tate pairing, which is computationally more efficient than the Weil pairing. The Tate pairing is an example of a bilinear map of the form e: $\mathbb{G}_0 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ where $\mathbb{G}_0$ and $\mathbb{G}_2$ are groups of order r. We will also make use of a complexity assumption called the co-Bilinear Diffie Hellman assumption (co-BDH), this assumption states:

Given P, xP, $\in \mathbb{G}_0$ and yQ, $zQ \in \mathbb{G}_1$ for unknown random values x, y and z, no polynomial time algorithm can compute $e(P,Q)^{xyz}$ with non-negligible probability.

2.1 Properties of the Tate pairing over this curve.

A curve with security multiplier $k = 2$ defined over the field $\mathbb{F}_{q^2}$ contains $r + 1$ subgroups, each containing r points and sharing the point at infinity O. In the following notation we are only concerned with two of these groups, which we refer to as $\mathbb{G}_0$ and $\mathbb{G}_1$. In implementation it is convenient to make one of these the group of points of order r defined over the field $\mathbb{F}_q$. In the following notation we will refer to points using capital letters, with different letters referring to different points. All points will be marked with a subscript to denote which group they below to, elements of group $\mathbb{G}_0$ will be subscripted with a 0, whilst elements of $\mathbb{G}_1$ will be subscripted with a 1. Results of the pairing operation are elements of the group $\mathbb{G}_2$. Therefore $P_0, Q_0, R_0 \in \mathbb{G}_0$. $P_1, Q_1, R_1 \in \mathbb{G}_1$.

- $e(P_0, Q_0) = e(P_1, Q_1) = 1$
  for any two points in the same subgroup.
- $e(P_0, Q_1) \neq 1$
  for any two points from different subgroups.
- $e(P_0, Q_1) \neq e(Q_1, P_0)$
  for any two points from different subgroups.
- $e(xP_0, yQ_1)^z = e(P_0, Q_1)^{xyz}$
  bilinearity
- $e(x(Q_1 + P_0), yQ_1) = e(xQ_1, yQ_1) \times e(xP_0, yQ_1) = 1 \times e(xP_0, yQ_1) = e(xP_0, yQ_1)$
  pairing properties hold for a third point that results as the addition of points in the two separate subgroups.

3 The Proposed Secret Handshake Scheme

As with all identity based key agreement schemes the proposed secret handshake scheme consists of the following 3 algorithms

Setup: The Public Key Generation Centre (PKGC) generates a random master secret s, selects an appropriate elliptic curve. The PKGC then picks a generator point in the subgroup $\mathbb{G}_0$ called P_0, and calculates $P_{pub_0} = sP_0$. The PKGC then hashes identities and private keys to the second subgroup $\mathbb{G}_1$.

The Secret Groups Key Generation Centre (SGKGC) picks it's generator point in $\mathbb{G}_1$, say P_1, it then generates its own master secret x, and calculates $P_{pub_1} = xP_1$, and hashes identities and private keys on $\mathbb{G}_0$. The two systems are effectively a mirror image of each other, each having a unique master secret.

Extract: Identity strings as certified by the PKGC are hashed to elements of $\mathbb{G}_1$. Where a particular identity is hashed to ID_1, the corresponding private key is calculated as sID_1. The hashing algorithm is made known to the public so that public keys can be derived directly from the identity string. In a similar fashion, identities certified by the SGKGC are hashed to elements of $\mathbb{G}_0$, where an identity is hashed to ID_0, the private key is calculated as xID_0. It is assumed that everyone has a PKGC issued identity and private key, whilst only members of the secret group have SGKGC issued public keys. Importantly these secret group public keys are not distributed, such a list of public keys may imply membership of the secret group. Instead it is assumed that they are generated when needed, directly from the identity string, this can be the same identity as is used for generating PKGC public keys, and so need not be secret. It is assumed that the hash algorithm needed to do this is made known to the members of the group.

Handshake/Key-Agreement: There are three possible scenario's for the handshake phase

1. Both parties are not members of the secret group.
2. One party is a member of the secret group.
3. Both parties are members of the secret group.

We will look at these in turn.

3.1 Both parties are not members of the secret group

The algorithm uses a hash function $H(.) : R \in \mathbb{G}_2 \rightarrow \{0,1\}^k$, that is, a one way maps from an element on the field over $\mathbb{F}_{q^k}$ to a suitable asymmetric session key. When both parties are not members of the group there is obviously no secret handshake and the protocol is simply Smarts' key agreement protocol[10]. To recap, this is as follows:

Alice's public identity is A_1, Bob's public identity is B_1

Alice generates a random number a and calculates aP_0.
Alice then sends this to Bob.

Alice $\rightarrow$ Bob

- aP_0

Bob generates a random number b and calculates bP_0.
Bob then sends this to Alice.

Bob $\rightarrow$ Alice

- bP_0

Alice calculates her shared secret as

- $H(e(sA_1, bP_0) \times e(B_1, aP_{pub_0})) = H(e(bA_1 + aB_1, P_{pub_0}))$

Likewise, Bob calculates his shared secret as

- $H(e(sB_1, aP_0) \times e(A_1, bP_{pub_0})) = H(e(bA_1 + aB_1, P_{pub_0}))$

And so they have a common shared secret.

3.2 One party is a member of the secret group

When one person is a member of the group it is important that the other person is unable to distinguish between this instance of the protocol and an instance of the protocol as outlined in section 3.1 above. Obviously they cannot be exactly the same, or another member of the group will not be able to distinguish them either.

Lets say that Alice is now a member of a group of which Bob is not a member.

Alice generates two random numbers, a and a'. One for her public group shared secret and the other for her secret group shared secret. Alice then sends the following value off to Bob.

Alice $\rightarrow$ Bob

- $aP_0 + a'P_1$

Bob, not being a member of the group, thinks that he is engaging in the protocol as outlined in section 3.1 and so responds as he did previously.

Bob $\rightarrow$ Alice:

- bP_0

Using the Tate pairing Alice can pair the value that she has just received from Bob with the point P_0. If the result is 1 then Bob is not in the group, since as noted above $e(xP_0, yP_0) = 1$, otherwise he is. Alice establishes that Bob is not in the group and so completes the protocol as if she was not in the group.

Alice calculates her shared secret as

- $H(e(sA_1, bP_0)) \times e(B_1, aP_{pub_0}) = H(e(bA_1 + aB_1, P_{pub_0}))$

Likewise, Bob calculates his shared secret as

- $H(e(sB_1, aP_0 + a'P_1) \times e(A_1, bP_{pub_0})) =$
  $H(e(sB_1, aP_0) \times e(sB_1, a'P_1) \times e(A_1, bP_{pub_0})) =$
  $H(e(sB_1, aP_0) \times 1 \times e(A_1, bP_{pub_0})) =$
  $H(e(bA_1 + aB_1, P_{pub_0}))$

And so they have the common shared secret.

3.3 Both parties are members of the group

When both parties are members of the group, they perform the secret handshake. This is done as follows.
Obviously neither party knows before the handshake whether the other is in the secret group. Therefore they have to perform a handshake that allows them to obtain a shared public secret as well.

Alice's secret group identity is A_0, Bob's secret group identity is B_0. Note, importantly, these points are in $\mathbb{G}_0$whereas their public identities A_1 and B_1 are in $\mathbb{G}_1$. It is assumed that all members of the secret group have the necessary algorithms to hash from an identity string to an element of the group $\mathbb{G}_0$, and therefore that explicit public keys are not kept in a list anywhere, but generated on the fly. Also members of the secret group Alice and Bob are equiped with the private keys xA_0 and xB_0respectively

Alice generates her two random numbers, a and a' again. Alice then sends the following value off to Bob.

Alice → Bob

- $aP_0 + a'P_1$

Bob, now being a member of the group, replies to Alice in a similar fashion.

Bob → Alice:

- $bP_0 + b'P_1$

Alice and Bob both now pair the value that they have been given with the point P_0. They see that the result is non-trivial, since, as mentioned above $e(P_0, P_0 + P_1) = e(P_0, P_1) \neq 1$ and so realise that they are both in the secret group.

They both now, instead of using their public identities, use their secret group identities and perform the usual handshake. Alice generates her shared secret value as

- $H(e(xA_0, bP_0 + b'P_1) \times e(B_0, a'P_{pub_1})) =$
  $H(e(xA_0, b'P_1) \times e(B_0, a'P_{pub_1})) =$
  $H(e(b'A_0 + a'B_0, P_{pub_1}))$

Likewise, Bob calculates his shared secret as

- $H(e(xB_0, aP_0 + a'P_1)) \times e(P, b'P_{pub_1}) =$
  $H(e(xB_0, a'P_1)) \times e(P, b'P_{pub_1}) =$
  $H(e(b'A_0 + a'B_0, P_{pub_1}))$

And so they have a shared secret that is unique to them as members of the secret group.

4 The Proposed Signcryption Scheme

Signcryption - first proposed by Zheng [11] - is the combination of encryption and signing operations in a way that is more efficient than doing both operations desperately. We now propose a signcryption scheme for low bandwidth channels based on the above technique of adding points in different subgroups. In practical implementation this protocol saves 513 bits for a reasonably secure system with $k = 2$, when compared with the bandwidth of the Malone-Lee Identity Based Signcryption system[12]. It does this by removing one of the points defined over $\mathbb{F}_q$ that was previously sent across as part of the signcryption (512bits for the x co-ordinate of the point, 1 bit for the y co-ordinate). Depending on the curves used it also makes bandwidth savings over the schemes of Libert & Quisquater when using the Tate pairing. [13]

The protocol uses hash functions $H_1 : R \in \mathbb{G}_2 \rightarrow \{0,1\}^k$ and $H_2 : \{0,1\}^* \rightarrow \{0,1\}^k$. It also uses a suitable symmetric encryption algorithm, keyed with a session key, denoted $SE_{sessionkey}(.)$.

The protocol is composed of four steps. Setup and Extract, carried out by the Key Generation Centre, and Signcryption and Unsigncryption, carried out by end users. To allow for non-repudiation services, if the receiver wishes to prove the origin of a message to a third party then he can decrypt the message and the signature must hold.

Setup: The KGC generates a random value s. The KGC publishes a group generator for the first subgroup P_0, and the point $P_{pub_0} = sP_0$, keeping s a secret.

Extract: The KGC generates each users private key, first by hashing their identity to a point on the $\mathbb{G}_1$ to generate their public key and then computing the private key by muiltiplying this point by s. It is assumed that the hash algorithm used to generate public keys from identity strings is publicly known.

Signcryption: Alice has as her public key A_1 and her corresponding private key is sA_1. She generates a random number r and calculates rP_0. Bob has as his public key B_1 and his corresponding private key is sB_1. The signcryption scheme proceeds as follows:

Alice calculates the following:

- $sk = H_1(e(P_{pub_0}, B_1)^r)$
- $v = H_2(H_2(message) \oplus sk)$
- $K = vsA_1 + rP_0$
- $cipher = SE_{sk}(message)$

Alice $\rightarrow$ Bob

- $Signcryption = (K, cipher)$

We note here that the signcryption consists of just one point, and a small piece of cipher text.

Unsigncryption: Bob checks the validity of the signcryption as follows:

Bob calculates:

- $sk' = H_1(e(K, sB_1)) =$
  $H_1(e(vsA_1, sB_1) \times e(rP_0, sB_1)) =$
  $H_1(e(P_0, sB_1)^r)$
- $message' = SE_{sk'}(cipher)$
- $v' = H_2(H_2(message') \oplus sk')$
- Accept iff $e(K, P_0) = e(A_1, P_{pub_0})^{v'}$

To see that this is the case consider
If $message' = message$ and $sk' = sk$, then $v' = v$.

$e(K, P) =$
$e((vsA_1 + rP_0), P_0) =$
$e(vsA_1, P_0) \times e(rP_0, P_0) =$
$e(vsA_1, P_0) \times 1 =$
$e(vsA_1, P_0) =$
$e(vA_1, sP_0) =$
$e(A_1, P_{pub_0})^{v'}$

5 Security of the Proposed Schemes

5.1 Security of the Secret Handshake

We now analysis the security of the proposed schemes, looking at the security of the secret handshake scheme first, taking as our reference the accepted criteria for an Authenticated Key Agreement with Key Confirmation (AKC). We use the notation of Blake et al. [14].

A protocol is a secure AKC protocol if:

- *In the presence of a benign adversary on* $\prod_{i,j}^{s}$ *and* $\prod_{i,j}^{t}$*, both oracles (parties to the key agreement) always accept holding the same session key, FK, and that this key is uniformly distributed at random on* $\{0,1\}^k$

And if for every adversary E

- *If uncorrupted oracles* $\prod_{i,j}^{s}$ *and* $\prod_{i,j}^{t}$ *have matching conversations then both oracles accept and hold the same session key FK.*
- *The probability of No-Matching*^{E}*is negligible.*
- *advantage*^{E}*(K) is negligible*

The first condition says that in the presence of a benign adversary, oracles always accept holding the same, randomly distributed key. The second says that in the presence of any adversary if two entities behave correctly, and the transmissions between them are not tampered with, then both accept and hold the

same key. The third says that essentially the only way for any adversary to get an uncorrupted entity to accept in a run of the protocol with any other uncorrupted entity is by relaying communications like a wire. The fourth says that no adversary can learn any information about a session key held by a fresh oracle. [14]

Theorem 1. In the random oracle model, the proposed protocol is an AKC protocol.

Proof: The protocol satisfies all of the conditions outlined above.

- Both oracles accept and hold the same session key as a result of the bilinearity of the pairing.
- The session key is uniformly distributed over $\{0,1\}^k$ as a result of using a random oracle on the output of the pairing. Also if even one party to the protocol picks their random element truly at random then the result is random.
- Condition two above follows from condition one.
- Conditions three and four will be dealt with simultaneously. Consider an algorithm E that has $advantage^E(K)$ with non-negligible probability. This can be transformed by another algorithm F to non-negligible advantage in breaking the Bilinear Diffie Hellman Assumption.

F has as its task to gain non-negligible advantage in breaking the co-Bilinear Diffie Hellman assumption:

Given $P_0, xP_0 \in \mathbb{G}_0$ and yQ_1 and $zQ_1 \in \mathbb{G}_1$ for unknown values x, y, z, compute $e(P_0, Q_1)^{xyz}$.

E can with non-negligible advantage, predict whether a random sequence $R = \{0,1\}^k = H(e(bA_1 + aB_1, P_{pub_0}))$, having seen aP_0 and bP_0. Note that in the protocol the value $aP_0 + a'P_1$ is sent across for the secret handshake, but since one point, for example $a'P_1$ disappears in the pairing the complexity assumption rests on aP_0. Therefore we will refer to $aP_0 + a'P_1$ simply as aP_0. Since H is a true random oracle there is no other way to guess R with non-negligible advantage other than to be able to guess $e(bA_1 + aB_1, P_{pub_0})$ with non-negligible advantage.

In the random oracle model we can formulate, for random i and j, values for P_0(representing the group generator), xP_0 (representing the KGC's public point), yQ_1 (representing Alice's public point), $(z \times (i+j)^{-1})Q_1$ (representing Bob's public point), $(i \times y)P_0$ (representing Alice's token to Bob), $(j(i+j)^{-1} \times z)P_0$ (representing Bob's token to Alice).

Remember that the random oracle's do not know the private keys for Alice and Bob, nor the server secret value s, but are allowed a best guess. Computing $e(bA_1 + aB_1, P_{pub_0}) = e(bP_0, sA_1) \times e(sP_0, aB_1) = e((zj(i+j)^{-1})P_0, yQ_1) \times e(xP_0, (z(i+j))Q_1) = e(P_0, Q_1)^{xyz}$. Since the oracles do not know the real server secret, it can be seen that non-negligible $advantage^E(K)$ in guessing a genuine fresh shared secret, or the ability to corrupt the communication in a way that an uncorrupted oracle will still accept, can be transformed into non-negligible

advantage in solving the co-BDH assumption. The full example of this proof is explained by Chen and Kulda using the BDH assumption. [15]

The protocol also has the following properties:

1. *Forward Secrecy:* Since each run of the protocol interactively uses random secrets, destruction of these random secrets after a run of the protocol means that even if the long term private keys are compromised past session keys will not be compromised. Entities can protect themselves by doing this.
2. *High Key EntrEntopyopy:* Provided that at least one of the entities has a good source of randomness the agreed session key will be random. Entities can protect themselves by maintaining a good source of randomness for themselves.
3. *Known Session Keys:* Because each key generation is randomised, knowledge of previously agreed session keys does not reveal any information about a freshly generated session key.

5.2 Security of the proposed Signcryption scheme

We use as our security model an idea proposed by John Malone-Lee. This is a two part definition in which he looks at the encryption and unforgeability properties of the signcryption scheme separately. This security model for encryption is called indistinguishability of identity-based signcryptions under chosen ciphertext attack (IND-ISC-CCA) and is a natural adaptation of the de facto standard for public key encryption schemes: indistinguishability of encryptions under chosen ciphertext attack.[12]

We recap:

Definition 1. We say that an identity based signcryption scheme (IDSC) has the indistinguishability against adaptive chosen ciphertext attacks property (IND-IDSC-CCA) if no polynomially bounded adversary has a non-negligeable advantage in the following game.

- The challenger runs the Setup algorithm with a security parameter k and sends the system parameters to the adversary.
- The adversary A performs a polynomially bounded number of requests:
- Signcryption request: A produces two identities A, B and a plaintext m. The challenger computes $sA = Keygen(A)$ and then $Signcrypt(m, sA, B)$ and sends the result to A.
- Unsigncryption request: A produces two identities A and B , a ciphertext σ. The challenger generates the private key $sA = Keygen(A)$ and sends the result of $Unsigncrypt(\sigma, sA, B)$ to A (this result can be the $\perp$ symbol if σ is an invalid ciphertext).
- Key extraction request: A produces an identity A and receives the extracted private key $sA = Keygen(A)$. A can present its requests adaptively: every request may depend on the answer to the previous ones.

- A chooses two plaintexts m_0 and m_1, and two identities A and B on which he wishes to be challenged. He cannot have asked the private key corresponding to A nor B in the first stage.
- The challenger takes a random bit b and computes $C = Signcrypt(m_b, sA, B)$ which is sent to A.
- A asks again a polynomially bounded number of requests just like in the first stage. This time, he cannot make a key extraction request on A nor B and he cannot ask the plaintext corresponding to C.
- Finally, A produces a bit b_0 and wins the game if $b_0 = b$.
- The adversarys advantage is defined to be $Adv(A) = |\text{prob(win) - } 1/2|$

Definition2. An identity based signcryption scheme (IDSC) is said to be secure against an existential forgery for adaptive chosen messages attacks (EF-IDSC-ACMA) if no polynomially bounded adversary has a non-negligeable advantage in the following game

- The challenger runs the Setup algorithm with a security parameter k and gives the system parameters to the adversary.
- The adversary A performs a polynomially bounded number of requests just like in the previous definition.
- Finally, A produces a new triple (σ, A, B) (i.e. a triple that was not produced by the signcryption oracle), where the private key of A was not asked in the second stage and wins the game if the result of $Unsigncrypt(\sigma, sA, B)$ is not the $\perp$ symbol.
- The adversarys advantage is simply its probability of victory.
- In this definition, the adversary is allowed to ask the private key corresponding to the identity B for which the ciphertext he produces must be valid. This condition is necessary to obtain the non-repudiation property and to prevent a dishonest recipient from sending a ciphertext to himself on Alices behalf and trying to convince a third party that Alice was the sender.

Proof of IND-IDSC-CCA security of our signcryption scheme First we consider the indistinguishablity of cipher texts produced by this signcryption scheme. Again, we show that an adversary that has non-negligible advantage in distinguishing ciphertexts against this signcryption scheme has non-negliglable advantage advantage against co-BDH assumption. This is a very straightforward transformation. The known points are B_1, where this is Bob's public key, rP_0, where r is the random mask, and sP_0, for the system wide secret s. Because the result of the pairing $e(B_1, sP_0)^r$ is hashed using a true random oracle an attacker can only distinguish between ciphertexts if they can calculate, with non-negligable probability $e(B_1, sP_0)^r$, knowing only rP_0, B_1 and sP_0. This is the co-Bilinear Diffie Hellman problem. We also note that the attack outlined in Libert & Quisquater [13], that is distinguishing between ciphertexts just by checking signatures, would not succeed as the signature is not directly visible in the signcryption. This is accomplished by the hashing of the message and the secret session key together in the signature, and the use of a symmetric encryption algorithm to generate the cipher text.

Proof of EF-ISC-ACMA security of our signcryption scheme We now consider the strength of our signcryption in the face EF-ISC-ACMA challenge, that is, its resistance to forgery. We can use the method of Hess [16], using the forking lemma of Pointcheval and Stern [17] to show that the ability to produce two ciphertexts in the random oracle model implies a non-negligible advantage in Computational Diffie Hellman Assumption. We assume that we have generated two forged signatures on the same message, using different outputs of the hash on the same message, all imputs are kept the same.

Forgery 1:

- $v = H_2(H_2(message) \oplus sk)$
- $K = vsA_1 + xP_0$

Forgery 2:

- $v' = H_2(H_2(message) \oplus sk)$
- $K' = vsA_1 + xP_0$

Attack:

- $K - K' = (vsA_1 + xP_0) - (v'sA_1 + xP_0) = (v - v')sA_1$
- $i = (1/(v - v'))$
- $sA_1 = i(K - K')$

The Computational Diffie Hellman Assumption states that, knowing $P, Q \in \mathbb{G}_1$ and $R = sP$, it is not computationally feasible to calculate sQ [18]. If this was trivial, it would be possible to generate private keys at will, just from knowing the system parameters.

6 Conclusion

We have presented two new protocols from the Tate pairing. We have seen how it is possible to introduce a covert channel into Smart's Handshake, such that it is possible to generate two concurrent secrets from the one instance of the protocol, we call the second of these secrets the secret handshake, since it requires additional knowledge to compute. We have also shown an efficient signcryption scheme in which just one point and one small piece of cipher text is transmitted.

References

1. A. Shamir, "Identity based cryptosystems and signature schemes," in *Proceedings of Crypto '84 conference*, 1984.
2. C. Cocks, "An identity based encryption scheme based on quadratic residues," 2001, "http://www.cesg.gov.uk/site/ast/idpkc/media/ciren.pdf.
3. D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," 2001, http://citeseer.nj.nec.com/boneh01identitybased.html.
4. K. Paterson, "Id-based signatures from pairings on elliptic curves," Cryptology ePrint Archive, Report 2002/004.
5. J. Cha and J. Cheon, "An identity-based signature from gap diffie-hellman groups," Cryptology ePrint Archive, Report 2002/018.
6. C. Gentry and A. Silverberg, "Hierarchical id-based cryptography," Cryptology ePrint Archive, Report 2002/056.
7. F. Zhang and K. Kim, "Id-based blind signature and ring signature from pairings," 2002, http://citeseer.nj.nec.com/article/zhang02idbased.html.
8. Balfanz, Durfee, Narendar, Smetters, Staddon, and Wong, "Secret handshakes from pairing-based key agreements," *IEEE*, 2003, http://www2.parc.com/csl/members/balfanz/publications/handshakes.pdf.
9. M. Scott, "The tate pairing," 2003, http://www.computing.dcu.ie/~mike/tate.html.
10. N. Smart, "An identity based authenticated key agreement protocol based on the weil pairing," 2001. [Online]. Available: http://eprint.iacr.org/2001/111/
11. Y. Zheng, "Digital signcryption or how to achieve cost (signature & encryption) << cost(signature) + cost(encryption)," *Lecture Notes in Computer Science*, vol. 1294, pp. 165–179, 1997, http://citeseer.nj.nec.com/zheng97digital.html.
12. J. Malone-Lee, "Identity-based signcryption," Cryptology ePrint Archive, Report 2002/098, 2002. [Online]. Available: \url{http://eprint.iacr.org/2002/098/}
13. B. Libert and J. Quisquater, "New identity based signcryption schemes from pairings," Cryptology ePrint Archive, Report 2003/023, 2003, http://eprint.iacr.org/2003/023/.
14. J. D. Blake-Wilson S. and M. A., "Key agreement protocols and their security analysis." http://citeseer.nj.nec.com/53608.html.
15. L. Chen and C. Kudla, "Identity based authenticated key agreement from pairings," Cryptology ePrint Archive, Report 2002/184, 2002, http://eprint.iacr.org/2002/184/.
16. F. Hess, "Exponent group signature schemes and efficient identity based signature schemes based on pairings," Cryptology ePrint Archive, Report 2002/012.
17. D. Pointcheval and J. Stern, "Security arguments for digital signatures and blind signatures," *Journal of Cryptology: the journal of the International Association for Cryptologic Research*, vol. 13, no. 3, pp. 361–396, 2000, http://citeseer.nj.nec.com/pointcheval00security.html.
18. Y. Yacobi, "A note on the bilinear diffie-hellman assumption," Cryptology ePrint Archive, Report 2002/113, 2002, http://eprint.iacr.org/2002/113.