AN IDENTITY BASED AUTHENTICATED KEY AGREEMENT PROTOCOL BASED ON THE WEIL PAIRING

N.P. SMART

ABSTRACT. We describe an ID based authenticated two pass key agreement protocol which makes use of the Weil pairing. The protocol is described and its properties are discussed including the ability to add key confirmation.

1. INTRODUCTION

Key agreement is one of the fundamental cryptographic primitives after encryption and digital signatures. The first modern protocol for key agreement was the Diffie-Hellman protocol given in their seminal paper [4], however this protocol needs to be used with care. Over the years a number of security properties have been seen to be important in key agreement protocols. For example the basic Diffie-Hellman protocols suffers from the man-in-the-middle attack because it does not attempt to authenticate the communicating parties. A simple solution would be to combine a key agreement protocol with a digital signature scheme to obtain an *authenticated key agreement* protocol (or AK protocol in the language of [7]). This solution has a number of problems, not least of which is the fact that the message lengths are now much greater than in the standard Diffie-Hellman protocol.

In [7] Law, Menezes, Qu, Solinas and Vanstone propose an AK protocol, often called the MQV protocol, which provides authentication of the parties but with message flows identical to the message flows in naive Diffie-Hellman. Hence, the authentication is achieved without an increase in bandwidth and without an increase in the number of message flows. In addition the MQV protocol can easily be transformed into a three flow protocol which provides the additional property of *key confirmation*, such a protocol being called a *authenticated key agreement protocol with key confirmation*, or AKC protocol for short.

The MQV protocol works by assuming each entity has a static public/private Diffie-Hellman key pair, and that each other entity knows the public key of each other entity. When a session key wishes to be determined a pair of ephemeral Diffie-Hellman public keys are exchanged. The ephemeral and static keys are then combined in an ingenious way so as to obtain the agreed session keys. One should note that the problem of authenticating the session key is replaced by the problem of authenticating the static public keys. But this latter problem can be solved using a traditional approach based on a *public key infrastructure* (PKI).

Recently, Cocks [3] and Boneh and Franklin [1] have proposed two identity based encryption schemes which potentially allow the replacement of a PKI with a system

Key words and phrases. cryptography, ID-based cryptography, key agreement, elliptic curve cryptography.

where ones identity becomes the public key and a key generation centre helps generate users private keys. The system of Cocks is based on the Quadratic Residuosity problem, whilst that of Boneh and Franklin relies on the Weil pairing.

In this paper we describe a two pass identity based authenticated key agreement protocol. Our protocol is based on the Weil pairing and combines the ideas of Boneh and Franklin with the tripartite Diffie-Hellman protocol of Joux [6]. The message flows in our new protocol are identical with the message flows of the two pass elliptic curve based unauthenticated Diffie-Hellman protocol, hence from the outside it looks like a Diffie-Hellman or MQV protocol. But the way the session key is produced will make use of the Weil pairing and the identity based static public keys.

Our protocol being ID based requires a trusted key generation centre. The protocol has the novel property that the key generation centre is able to recover the agreed session keys from the message flows and its secret key. Combined with a secret sharing scheme for the key generation centres secret key, this allows for an efficient ID based escrow facility for sessions. This would enable law enforement agencies to decrypt messages encrypted with the session keys, after having obtained the necessary warrants. Note, that the ID based encryption scheme of Boneh and Franklin [1], also has the ability for the key generation centre to decrypt messages at will.

2. The Weil Pairing

In this section we shall summarize the properties we require of the Weil pairing, much of the details can be found in [1], [8] and [10]. We let $\mathbb{G}$ denote a prime order subgroup of an elliptic curve E over the field $\mathbb{F}_q$. Let the order of $\mathbb{G}$ be denoted by l and define k to be the smallest integer such that

$$l | q^k - 1.$$

In practical implementations we will require k to be small and so we will usually take E to be a supersingular curve over $\mathbb{F}_q$.

The modified Weil pairing is a map

$$\hat{e} : \mathbb{G} \times G \to \mathbb{F}_{q^k}^*,$$

which satisfies the following properties

1. Bilinear:
   - $\hat{e}(P_1 + P_2, Q) = \hat{e}(P_1, Q) \cdot \hat{e}(P_2, Q)$.
   - $\hat{e}(P, Q_1 + Q_2) = \hat{e}(P, Q_1) \cdot \hat{e}(P, Q_2)$.
2. Non-degenerate: There exists a $P \in \mathbb{G}$ such that $\hat{e}(P, P) \neq 1$.
3. Computable : One can compute $\hat{e}(P, Q)$ in polynomial time.

The non-degeneracy, as we have defined it, does not hold for the standard Weil pairing $e(P, Q)$, but it does hold for the modified Weil pairing $\hat{e}(P, Q)$. That the Weil pairing is efficiently computable follows from an unpublished, but much referenced, algorithm of Miller [9]. The modified Weil pairing that we use is defined in [1] and it is computed in exactly the same way by using Miller's algorithm.

Originally the existence of the Weil pairing was thought to be a bad thing in cryptography. For example in [8] it was shown that the discrete logarithm problem in supersingular curves was reducible to that in a finite field using the Weil pairing. This led supersingular elliptic curves to be dropped from cryptographic use. This situation changed with the work of Joux [6], who gave a simple tripartite Diffie-Hellman protocol based on the Weil pairing on supersingular curves. Since Joux's paper a number of other applications have arisen, including an identity based encryption scheme [1] and a signature algorithm [2]. The extension to higher genus curves has also recently been fully explored in [5].

3. The AK and AKC Protocols

Suppose we have a subgroup $\mathbb{G}$ of an elliptic curve for which the modified Weil pairing $\hat{e}$ maps into the finite field $\mathbb{F}_{q^k}$. We assume that q^k is large enough to make solving discrete logarithms in the finite field infeasible and we assume that the elliptic curve contains a large prime subgroup of order l, such that solving discrete logarithms in the subgroup of order l is also infeasible.

Let

$$V : \mathbb{F}_{q^k}^* \longrightarrow \{0,1\}^*$$

denote a key derivation function. We shall not discuss the properties of this function in this paper, but simply note that such functions can be readily found in a number of standards documents. We let

$$H : \{0,1\}^* \longrightarrow \mathbb{G}$$

denote a cryptographic hash function. A simple definition for H would be to apply a standard cryptographic hash function to obtain a seed X and then to compute

$$X_i = X + i$$

for $i = 0, 1, 2, \ldots$ until a valid x-coordinate of a point was reached. The element in $\mathbb{G}$ is then defined to be a point with this x-coordinate. The exact choice of y-coordinate also needs to be fixed but this is easily done, for example in the case of characteristic greater than three one could select the value of the y-coordinate to be even.

3.1. **System Setup.**

The key generation center chooses a secret key

$$s \in \{1, \ldots, l-1\}.$$

The key generation centre produces a random $P \in \mathbb{G}$ and computes

$$P_{KGS} = [s]P.$$

Then the key generation centre publishes

$$(P, P_{KGS}).$$

When a user with identity ID wishes to obtain a public/private key pair, the public key is given by

$$Q_{ID} = H(ID).$$

The key generation centre computes the associated private key via

$$S_{ID} = [s]Q_{ID}.$$

Note, that this calculation can be performed using a distributed key generation centre using standard secret sharing methods.

3.2. **Authenticated Key Exchange.** Suppose two users A and B wish to agree a key. We denote the private keys of these uses by

$$S_A = [s]Q_A \text{ and } S_B = [s]Q_B,$$

which have been obtained from the key generation centre.

Each user generates an ephemeral private key, say a and b. The data flows are then the values of the corresponding ephemeral public keys $T_A = [a]P$ and $T_B = [b]P$, as the following diagram shows

$$\begin{array}{ccc} \text{User A} & & \text{User B} \\ a & & b \\ T_A = [a]P & \longrightarrow & T_A \\ T_B & \longleftarrow & T_B = [b]P \end{array}$$

User A then computes

$$k_A = \hat{e}([a]Q_B, P_{KGS}) \cdot \hat{e}(S_A, T_B)$$

and user B computes

$$k_B = \hat{e}([b]Q_A, P_{KGS}) \cdot \hat{e}(S_B, T_A).$$

The secret key is then

$$K = V(k_A) = V(k_B).$$

We first show that the secret shared keys agree,

$$\begin{aligned} k_A &= \hat{e}([a]Q_B, P_{KGS}) \cdot \hat{e}(S_A, T_B) \\ &= \hat{e}(Q_B, P_{KGS})^a \cdot \hat{e}(S_A, T_B) \\ &= \hat{e}(Q_B, P)^{as} \cdot \hat{e}(Q_A, P)^{bs} \\ &= \hat{e}(S_B, T_A) \cdot \hat{e}(Q_A, P_{KGS})^b \\ &= \hat{e}(S_B, T_A) \cdot \hat{e}([b]Q_A, P_{KGS}) \\ &= \hat{e}([b]Q_A, P_{KGS}) \cdot \hat{e}(S_B, T_A) \\ &= k_B. \end{aligned}$$

Note, that we have the equation

$$k_A = \hat{e}([a]Q_B + [b]Q_A, [s]P)$$

hence the shared secret depends on the identities Q_A, Q_B of the two parties, the secret key s of the key generation centre and the two ephemeral keys a, b.

3.3. **Efficiency.** The above protocol is *role symmetric*, in that both parties execute the same operations. It is easy to see that the above protocol requires each party to perform two elliptic curve point multiplications and two evaluations of the Weil pairing. One should compare this with the MQV protocol [7], which is also role symmetric. The MQV protocol requires each party to perform two full elliptic curve point multiplications plus one "half" one. Since evaluating the Weil pairing is a more costly operation than a "half" of a point multiplication one can conclude that MQV is more efficient than the above protocol.

However, MQV requires a deployed PKI so as to authenticate the long term public keys, whilst our system uses an identity based system. Hence, depending on the application domain it may be that our protocol is more applicable.

We note that in both MQV and our protocol the message flows, of the AK protocol, consist of one elliptic curve point. Hence, the bandwidth required by the two protocols are the same.

Finally we note that the above efficiency arguments assume that a similar sized elliptic curve is used in both instances. Hence, this probably means that for our protocol a supersingular elliptic curve over a field of characteristic three is to be prefferred.

3.4. **Security.** One can heuristically argue that the above protocol has the following security properties.

- *Known key security* : Each run produces a different session key, and knowledge of past session keys does not allow deduction of future session keys.
- *Forward Secrecy* : Compromising of a long term secret key, such as S_A, at some point in the future does not lead to the compromise of communications in the past. Note, however that compromise of the key generation centres long term secret s will allow anyone to compute the key via
$$\hat{e}(Q_B, T_A)^s \cdot \hat{e}(Q_A, T_B)^s.$$
This implies that the key generation centre is able to determine all secret session keys.
- *Key Control* : Neither party can control the outcome of the session key, by for example restricting it to lie in some predetermined small set.

3.5. **The Three Pass AKC Protocol.** Just as with the MQV protocol [7] it is trivial to add a key confirmation property to our protocol, making a three pass AKC protocol as follows: We require a message authentication code, MAC, and the key derivation function V now outputs two keys (k, k') which are the shared key and the MAC key respectively.

We let $R = \hat{e}([a]Q_B, P_{KGS}) = \hat{e}([b]Q_A, P_{KGS})$, the message flows then become

$$\begin{array}{ccc}
\text{User A} & & \text{User B} \\
a & & b \\
T_A = [a]P & \longrightarrow & T_A \\
(T_B, M_1) & \longleftarrow & \left\{ \begin{array}{l} T_B = [b]P, \\ M_1 = MAC_{k'}(2, B, A, R) \end{array} \right. \\
M_2 = MAC_{k'}(3, A, B, R) & \longrightarrow & M_2
\end{array}$$

The message authentication codes, M_1 and M_2, are checked by both parties. On the assumption that both parties choose a different ephmeral key for each run of the protocol one can heuristically argue that we will obtain the desired key confirmation.

4. Conclusion

We have proposed an ID based authenticated key agreement scheme which uses the Weil pairing. In addition we have shown how to add key confirmation to the basic protocol.

References

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

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

[3] C. Cocks. An identity based encryption scheme based on quadratic residues. To appear *Cryptography and Coding*, 2001.
[4] W. Diffie and M. Hellman. New directions in cryptography. *IEEE Trans. Info. Th.*, **22**, 644–654, 1976.
[5] S.D. Galbraith. Supersingular curves in cryptography. In *Advances in Cryptology - ASIACRYPT 2001*, Springer-Verlag LNCS 2248, 495-513, 2001.
[6] A. Joux. A one round protocol for tripartite Diffie-Hellman. In *Algorithmic Number Theory Symposium, ANTS-IV*, Springer-Verlag LNCS 1838, 385–394, 2000.
[7] L. Law, A.J. Menezes, M. Qu, J. Solinas and S. Vanstone. An efficient protocol for authenticated key agreement. To appear *Designs, Codes and Cryptography.*
[8] A.J. Menezes, T. Okamoto and S. Vanstone. Reducing elliptic curve logarithms to logarithms in a finite field. *IEEE Trans. Info. Th.*, **39**, 1639–1646, 1993.
[9] V. Miller. Short programs for functions on curves. Unpublished manuscript, 1986.
[10] J.H. Silverman. *The Arithmetic of Elliptic Curves.* GTM 106, Springer-Verlag, 1986.

Dept. Computer Science,, University of Bristol,, Merchant Venturers Building,, Woodland Road,, Bristol, BS8 1UB
E-mail address: nigel@cs.bris.ac.uk