Deniable Authenticated Key Establishment for Internet Protocols

Colin Boyd[1⋆] and Wenbo Mao[2⋆⋆] and Kenneth G. Paterson[3⋆⋆⋆]

[1] Information Security Research Centre, Queensland University of Technology, Brisbane, Australia
[2] Hewlett-Packard Laboratories, Filton Road, Stoke Gifford, Bristol BS12 6QZ, UK
[3] Information Security Group, Royal Holloway, University of London, Egham, Surrey TW20 0EX, UK

Abstract. We propose two public-key schemes to achieve "deniable authentication" for the Internet Key Exchange (IKE). Our protocols can be implemented using different concrete mechanisms and we discuss different options; in particular we suggest solutions based on elliptic curve pairings. The protocol designs use the modular construction method of Canetti and Krawczyk which provides the basis for a proof of security. Our schemes can, in some situations, be more efficient than existing IKE protocols as well as having stronger deniability properties.

1 Introduction

Systematic design of protocols for key establishment protocols was pioneered by Needham and Schroeder in 1978 [28]. Understanding of protocol design and security analysis has advanced steadily in the years following their ground-breaking work. This paper follows in that tradition.

In this paper we consider (i) a special privacy feature for Internet Protocols, (ii) its realisation and (iii) its security analysis. In this introduction section let us provide abstract-level descriptions of these three things.

1.1 Deniability for Internet Protocols

As electronic communications become ever more a part of everyday life, issues of individual privacy have started to increase in relevance. One property that was not often thought important in the past for key establishment, but is seen as increasingly relevant today, is *deniability*. This is a privacy property that ensures protocol participants can later deny taking part in a particular protocol run. Such a property has been declared as desirable for new protocols proposed to secure the IP (Internet Protocol) level on Internet communications [22]. A

⋆ Research conducted during a visit to Hewlett-Packard Laboratories, Bristol.
⋆⋆ Research partially funded by the EU Fifth Framework Project IST-2001-324467 "CASENET".
⋆⋆⋆ Supported by the Nuffield Foundation, NUF-NAL 02.

deniability service offered at the IP layer preserves this privacy feature from the upper layers, in particular, the application layer. This is important because sophisticated anonymity and privacy properties offered at the application layer can often be undermined by inappropriately designed lower layer protocols. Therefore, deniability at the IP layer should not be viewed as a luxury feature: it is an important option.

1.2 An Underlying Technique

Identity-based public-key cryptography (ID-PKC), in which public keys can be obtained and trusted without the use of certificates and without the need for a CA-based public-key infrastructure (PKI), has been the subject of much recent interest. This interest was sparked by three seminal papers being positive applications of a cryptanalysis result of Menezes, Okamoto and Vanstone [27], called the MOV reduction, which used a pairing technique to reduce the difficulty of the elliptic curve discrete logarithm problem in a "weak curve" group to that in a finite field. The first two were independent original works: one was by Sakai, Ohgishi and Kasahara [32] whose application of the MOV reduction is an ID-based, non-interactive key agreement protocol; the other was by Joux [23] whose application of the MOV reduction is a one-round three-party Diffie-Hellman key sharing (Joux names it "tripartite Diffie-Hellman"). Boneh and Franklin [6] further applied a "distortion map" technique of Verheul [34] which improved positive applications of the pairing technique: their result is the first proven secure and fully practical ID-based encryption scheme, answering a long-standing open problem of Shamir [33].

ID-PKC makes use of an alternative trust model to a traditional PKI, wherein a trusted authority (TA) is trusted to create and deliver private keys to entities, and not to abuse its knowledge of these keys. This trust model is well suited to building security solutions when a central TA can be used (for example, in many corporate scenarios). The schemes of [32, 6] will form the basis for our construction of alternatives to IKE which enjoy the strongest possible deniability properties.

1.3 Provable Security for Key Establishment Protocols

Today there are many protocols for key establishment that have a proof of security, particularly using the fairly mature model of Bellare and Rogaway [4, 5]. Yet new protocol designs continue to appear in the literature as new requirements are revealed for different applications. A feature of the Bellare-Rogaway model is that proofs tend to be monolithic and fragile; a small change in a protocol design may require a completely new proof to be developed. An alternative, newer, model by Canetti and Krawczyk [11] has the advantage that proofs are modular, making it easier to re-use partial proofs and instantiate abstract protocols with different concrete algorithms.

1.4 Our Work

The purpose of this paper is to suggest new, identity-based key establishment protocols that are suitable for use as an IKE replacement. We concentrate on two properties that have been lacking in some previous proposals, namely strong deniability and provable security. Our protocols enjoy stronger deniability properties than previous protocols based on signatures and a traditional PKI trust model. At the same time we are able to propose flexible, provably secure solutions through the use of the Canetti-Krawczyk design approach, and derive solutions which can be at least as efficient as other proposed protocols.

We regard the following as the main contributions of the paper.

- A generic design for identity-based key establishment with proven security in the Canetti-Krawczyk model.
- Two new abstract protocols with strong deniability properties.
- An investigation of different specific algorithms to provide concrete versions of our abstract protocols with properties comparable to existing proposals for Internet key exchange.

The remainder of this paper is structured as follows. In Section 2 we introduce IKE protocols and discuss one current proposal. Section 3 explores different ways to obtain shared secrets from public information, which is a main theme of our constructions. Section 4 explains how pairings on elliptic curves can be used to provide identity-based cryptography. This provides perhaps the most promising concrete implementation of our protocols. Then in Section 5 the provable security model of Canetti and Krawczyk is reviewed, including the building blocks that we need for our protocols. Section 6 puts the building blocks together to arrive at the new protocols with the properties we require.

2 Internet Key Exchange

The key establishment protocol known as Internet Key Exchange (IKE) was published by the IETF as a proposed standard for Internet Protocol Security (IPSec)[1] in 1998 [21]. However, many criticisms have been made on various aspects of the protocol, particularly its considerable complexity [17, 30, 35, 36]. IKE is really several protocols, since there are multiple options that can be used in any particular run. Recently there have been alternative proposals and it now seems inevitable that a new version of IKE will soon emerge with significant differences. Detailed descriptions of the current IKE protocols are given by both Borella [9] and Cheng [13] which are useful supplements to the description in the draft standard [21]. Canetti and Krawczyk [12] have provided a security proof for a generic protocol construction known as SIGMA [24]. This construction applies to the IKE protocol and so that is now supported by a formal security proof.

We use SIGMA as an illustration of a typical current IKE protocol. This also allows us to introduce our notation and give an idea of the main properties of

[1] We sometimes see IPsec in the literature, but we believe IPSec makes better sense.

IKE protocols. All the protocols in the current IKE versions use key agreement based on Diffie-Hellman key exchange [14], although the group in which this exchange takes place can be negotiated as part of the protocol.

Protocol 1 is based on generic Diffie-Hellman key exchange in a group of prime order q. Although we have used multiplicative notation we do not restrict the group, which could be a subgroup of $\mathbb{Z}_p^*$ or an elliptic curve group or any other appropriate choice. The protocol also makes use of any generic digital signature scheme which must be secure against adaptive chosen signature attacks – we denote the signature of entity X on message m as $Sig_X(m)$. A messages authentication code (MAC) is also required, and we denote the MAC of a message m using shared key K as $MAC_K(m)$. The shared secret is $Z_{AB} = g^{r_A r_B}$, and the session key is derived from Z_{AB}.

We note that this protocol, and many derivatives of it (including Σ_0, Σ_1 and their applications in IKE and IKEv2) suffer from a minor authentication flaw. We sketch this flaw in the Appendix.

A		B
$r_A \in_R \mathbb{Z}_q$		
$t_A = g^{r_A}$	$\xrightarrow{t_A}$	$r_B \in_R \mathbb{Z}_q$
		$t_B = g^{r_B}$
		$Z_{AB} = t_A^{r_B}$
		$K_m = H_1(Z_{AB})$
	$\xleftarrow{t_B, B, Sig_B(t_B, t_A), MAC_{K_m}(B)}$	
Verify signature		
$Z_{AB} = t_B^{r_A}$		
$K_m = H_1(Z_{AB})$		
Verify MAC		
	$\xrightarrow{A, Sig_A(t_A, t_B), MAC_{K_m}(A)}$	
		Verify MAC
		Verify signature

Protocol 1: Basic SIGMA protocol of Krawczyk

2.1 Deniability

One of the many desirable properties possessed by Protocol 1 is a form of anonymity which is called "plausible deniability". Harkins *et al.* [22] state that plausible deniability:

> ... enables Alice and Bob to communicate, leaving no ability for anyone to prove that they did have a conversation, even with the collusion of one of the parties.

Plausible deniability allows any user who may have taken part in a protocol run to deny that this was the case, since there is nothing to connect the user directly to a particular communication partner or protocol instance.

Plausible deniability may not provide strong enough anonymity for some situations. In particular, if the communications partners are mutually distrustful then they cannot be sure that the other will not provide a secret value which can link one or both partners to the protocol run. For example, although plausible deniability was a stated goal of Protocol 1, it has limitations in this regard. As long as both A and B cooperate, no third party can show that either of them was involved in a particular run. All that is available is a signature of both parties on two random Diffie-Hellman inputs. However, suppose that A wishes to implicate B in a particular session. By storing the random input, r_A, and the shared secret, Z_{AB}, A can show that B signed the parameters corresponding to session key. A similar situation holds if B wishes to implicate A.

A stronger form of deniability can be achieved using shared-key authentication. With a shared-key solution, either user in the protocol run could have produced all the messages in the run. An even stronger form of deniability can be obtained when the shared key is obtained using techniques from identity-based cryptography. Now, any user can simulate runs of the protocol involving *any* other potential user. Moreover, a user is not required even to reveal if he is registered with the trusted authority in order to take part in runs of the protocol; this is in contrast to schemes which employ certificates and a traditional PKI (including the signature-based schemes), where the existence of a certificate for a user indicates that user has registered, and identifies him as a *potential* protocol participant amongst all the users in a population. It is one aim of this paper to explore in detail specific protocols with these strong deniability properties..

3 Shared Secrets from Public Information

Below we consider key establishment protocols which can be used to derive a new key suitable to protect a subsequent communications session. The basis for the design is that the two users should be able to derive a shared secret which will be used as the key for a MAC. Because both parties can derive this secret, either of them can deny having taken part in the protocol. This property can be very useful in optimising the communications efficiency of protocols and can also be the basis of strong deniability. Suppose that F_{AB} is the fixed shared key between principals A and B used to authenticate the messages in a key establishment protocol. Even if one of A and B collaborates with an adversary, it is impossible to prove that the other was actually involved in the communication. This is because either party could have efficiently computed all the messages in the protocol run. Note that because F_{AB} is a static secret it is not suitable as a session key to protect communications by itself.

In order to ensure that the protocol is deniable, the shared secret should be derived from information that is, or could be, public. This means that no cryptographic channels need to be (in fact, can be) used by the parties before

the shared secret is derived. The public information could include several types of data. In particular it could consist of any set of the following items:

1. Identities of principals.
2. Public key parameters, such as moduli and group generators.
3. Public key values.
4. Public key certificates.

In this section we consider some available options for obtaining such shared secrets. We first look at more established methods and then introduce the newer identity-based methods. This latter option seems to have advantages in our scenario.

3.1 Shared Secrets from Public Keys

A well-known method to obtain a shared secret is the Diffie-Hellman key agreement protocol [14]. Diffie-Hellman keys are generally divided into two types: static keys and ephemeral keys. For static Diffie-Hellman each party has a fixed public key supported by a certificate; if x_A is Alice's private key then her public key is $y_A = g^{x_A}$, where g generates a suitable group. If Bob has a similar public/private key pair then he can generate the static Diffie-Hellman key $F_{AB} = y_A^{x_A} = g^{x_A x_B}$. Alice can also generate the same value in a symmetrical manner. The static Diffie-Hellman key is a shared secret which requires all four of the types of information enumerated above. In terms of deniability, a disadvantage of the use of certificates is that these are public values which can be used to verify that the owner of the certificate has at the least registered to take part in the scheme.

Girault's key agreement scheme [20] allows a shared secret to be obtained without exchange of messages, but it does require that the public keys of each party are known to the other. Girault's scheme uses *self-certified* public keys which cannot be used to verify that any user has registered in the scheme. The scheme uses an RSA modulus n and key pair e, d chosen by the trusted authority TA, together with an element g of high order in $\mathbb{Z}_n^*$. When A registers to use the scheme she chooses her private key x_A and provides g^{x_A} to TA, who calculates the self-certified public key $y_A = (g^{x_A} - ID_A)^d \bmod n$. (We have changed the sign of x_A from Girault's original description in order to maintain a uniform notation.) Girault's scheme uses the self-certified keys to produce an authenticated static shared key. If the public keys are already available this can be achieved with no message exchanges: A calculates $(y_B^e + ID_B)^{x_A}$ in order to produce $F_{AB} = g^{x_A x_B}$ and B calculates the same value in an analogous way.

Many identity-based schemes (including those using pairings explained below) require A's private key to be generated by the trusted authority. This need not be the case for Girault's scheme since S cannot find x_A from g^{x_A} without taking discrete logarithms in $\mathbb{Z}_n^*$. However, Saeednia [31] has pointed out that a malicious TA could choose n so that finding discrete logarithms is easy, and for this reason the size of n should preferably be at least 2048 bits. In addition S

should provide a proof that n is the product of two safe primes. This significantly reduces the bandwidth efficiency of Girault's scheme.

3.2 Shared Secrets from Identities

The notion of identity-based cryptography was first introduced by Shamir [33]. Shamir proposed a concrete identity-based signature scheme but no identity-based encryption scheme. Recently there has been a resurgence of interest in identity-based cryptography. The main reason was the discovery, beginning with Sakai, Ohgishi and Kasahara [32], and Boneh and Franklin [6], that identity-based cryptographic primitives can be realised using pairings on elliptic curves. A secondary reason was the realisation that ID-PKC uses a trust model that is more appropriate then a traditional CA-based PKI model for some types of application. A survey of recent activity in the area has been made by Paterson [29].

One striking consequence of an identity-based infrastructure is that any pair of parties can derive a shared secret without the need for any interaction or any certificates [32]. This seems to provide the optimal situation for deniability, since users have neither certificates nor public keys, and are able to derive the shared secret F_{AB} from knowledge of the peer's identity alone, along with any public parameters. The mathematical basis for this is pairings on elliptic curves, which allows new algebraic structures to be exploited. We outline how this works in the next section.

4 Pairings on Elliptic Curves

In this section, we review key agreement schemes and identity-based encryption schemes based on pairings. We use notation of Boneh and Franklin [7]. We let $\mathbb{G}_1$ be an additive group of a large prime order and $\mathbb{G}_2$ be a multiplicative group of the same order. We assume the existence of an efficiently computable bilinear map e from $\mathbb{G}_1 \times \mathbb{G}_1$ to $\mathbb{G}_2$. Typically, $\mathbb{G}_1$ will be a large prime-order subgroup of the group of points on an elliptic curve over a finite field, $\mathbb{G}_2$ will be the same order (multiplicative) subgroup of a related finite field (a small field extension) and the map e will be derived from either the Weil or Tate pairing on the elliptic curve. We assume that the pairing mapping has been modified (by applying Verheul's "distortion map" [7, 34]) and so for every non-infinity point $P \in \mathbb{G}_1$ it satisfies $e(P, P) \neq 1_{\mathbb{G}_2}$. By e being bilinear, we mean that for $Q, W, Z \in \mathbb{G}_1$, both

$$e(Q, W + Z) = e(Q, W) \cdot e(Q, Z) \quad \text{and} \quad e(Q + W, Z) = e(Q, Z) \cdot e(W, Z).$$

When $a \in \mathbb{Z}_q$ and $Q \in \mathbb{G}_1$, we write aQ for scalar multiplication of Q by a. As a consequence of bilinearity, we have that, for any $Q, W \in \mathbb{G}_1$ and $a, b \in \mathbb{Z}_q$:

$$e(aQ, bW) = e(Q, W)^{ab} = e(abQ, W).$$

We refer to the growing literature [1, 2, 6–8, 16, 18, 19] for a more comprehensive description of how these groups, pairings and other parameters should be selected in practice for efficiency and security. We assume in what follows that suitable groups $\mathbb{G}_1$ and $\mathbb{G}_2$, a map e and an element $P \in \mathbb{G}_1$ have been chosen, and that elements of $\mathbb{G}_1$ and $\mathbb{G}_2$ can be represented by bit strings of the appropriate lengths.

We also introduce here the computational problems that will form the basis of security for our identity-based schemes.

Bilinear Diffie-Hellman Problem (BDHP): Let $\mathbb{G}_1$, $\mathbb{G}_2$, P and e be as above with $\#\mathbb{G}_1 = \#G_2 = q$ being prime. The BDHP in $\langle \mathbb{G}_1, \mathbb{G}_2, e \rangle$ is as follows: Given $\langle P, aP, bP, cP \rangle$ with $a, b, c \in \mathbb{Z}_q^*$, compute $e(P, P)^{abc} \in \mathbb{G}_2$. An algorithm $\mathcal{A}$ has advantage ϵ in solving the BDHP in $\langle \mathbb{G}_1, \mathbb{G}_2, e \rangle$ if

$$\Pr\left[\mathcal{A}(\langle P, aP, bP, cP \rangle) = e(P, P)^{abc}\right] \geq \epsilon.$$

Here the probability is measured over random choices of a, b, c in $\mathbb{Z}_q^*$ and the internal random operation of $\mathcal{A}$. We assume that BDHP is a hard computational problem: letting q have the magnitude 2^k where k is a security parameter, there is no polynomial time (in k) algorithm which has a non-negligible advantage ϵ (again, in terms of k) in solving the BDHP for all sufficiently large k.

It is important for our later discussion to note that by appropriately building our groups from supersingular curves over fields of characteristic 3, we can arrange that elements of $\mathbb{G}_1$ have a representation that is 260 bits in size while, at the same time, the best known algorithm for solving BDHP has complexity at least on the order of 2^{80} (it is difficult to exactly quantify the security because it depends on the complexity of Coppersmith's discrete logarithm algorithm in the field $\mathbb{F}_{3^{163}}$). By generalising our definition of a pairing to allow two different groups on the left-hand side of the map e, we can have much greater flexibility in the choice of groups available and the size of their bit representations. We refer to [8, Section 4] and [2, 16] for further details.

Sakai *et al* [32] proposed the following non-interactive key agreement scheme using pairings (our description of their scheme uses the distortion mapped Weil pairing and so is a simplified one). Suppose A and B register ahead of time with a Trusted Authority (TA). The TA has chosen as a master secret an integer s, and gives A and B private keys which depend on s and the individual's identities. This is done as follows. The TA picks and makes public a cryptographic hash function H whose inputs are strings of arbitrary length and whose outputs are elements of $\mathbb{G}_1$. Then the TA computes as A's private key the value sQ_A where $Q_A = H(ID_A) \in \mathbb{G}_1$ is a publicly computable function of A's identity ID_A. Likewise, the TA gives B the value sQ_b where $Q_B = H(ID_B) \in \mathbb{G}_1$. Now A and B can both compute the shared key

$$F_{AB} = e(sQ_A, Q_B) = e(Q_A, Q_B)^s = e(Q_A, sQ_B),$$

each doing so by combining his private key with the other party's public identity information.

It is proven by Dupont and Enge [15] that a slight variant of this procedure generates a key which is secure against adversaries who can extract the private keys for arbitrary identities (except for A and B), provided that the BDHP is computationally infeasible. Notice however that the TA can generate A and B's common key for himself. So A and B must trust the TA not to produce communications protected by this key, nor to disclose the key to other parties.

The second new protocol that we propose in this paper makes use of public key encryption in addition to a MAC using the shared key F_{AB}. We want to employ identity-based encryption for this purpose in order to maintain enhanced anonymity properties. (Recall that a certificate-based scheme would immediately reduce the set of entities who could possibly have participated in a protocol to those who have been issued with a certificate.) The identity-based encryption scheme of Boneh and Franklin [6] using pairings is ideal for this purpose. It has the advantage that it can use the same supporting infrastructure as the key agreement scheme of [32]. Its security also rests on the difficulty of solving the BDHP, in an extension of the standard IND-CCA model suited to the identity-based setting. We refer to [6, 7] for the details of this scheme and its security properties.

5 Canetti and Krawczyk Approach to Provable Security

Mathematical techniques for protocol analysis have evolved over several years. One of the first and most influential was the logic of authentication of Burrows, Abadi and Needham [10] which was followed by many attempts to improve it. The success of the BAN logic has encouraged many other researchers to look for methods to gain assurance in the security of cryptographic protocols. These methods have come mainly from either the formal methods community or the cryptographic community.

In the cryptographic community, proofs for security have been based mainly on the notion of *reduction* from the problem of interest to a better understood computational problem. Such an approach to protocol analysis was initiated by Bellare and Rogaway in 1993 [4]. Although still popular today, one limitation of this approach is its monolithic nature. In 1998, Bellare, Canetti and Krawczyk [3] introduced a modular approach to protocol proofs, which was later refined by the last two of these authors [11]. In this section we give an informal overview of their approach, and describe the building blocks we use in our protocol design.

The model defines protocol principals who may run multiple sessions. A powerful adversary attempts to break the protocol by interacting with the principals. In addition to controlling all the communications the adversary is able to *corrupt* any principal and choose its long-term key (this models insider attacks). The adversary may also *reveal* any accepted session keys. The adversary must be efficient in the sense of being a probabilistic polynomial time algorithm.

Definition 1. *An AKE protocol is called* SK-secure *if the following hold.*

1. *If two uncorrupted parties complete matching sessions, then they both accept the same key.*

2. *Suppose a session key is agreed between two uncorrupted parties and has not been revealed by the adversary. Then the adversary cannot distinguish the key from a random string with probability significantly more than 1/2.*

Two adversarial models are defined. The first can be considered as an ideal world in which messages are authenticated magically. The second can be considered the real world in which we run our real protocols; this is the model in which we want to prove our protocols secure. However, in order to modularise the process protocols are first proven secure in the ideal world and then translated into the real world.

The authenticated-links adversarial model (AM) In this model the adversary is able to invoke protocol runs, masquerade as protocol principals, and find used session keys. Although the adversary is quite powerful it is unable to fabricate or replay messages which appear to come from uncorrupted parties.

The unauthenticated-links adversarial model (UM) In this model the adversary can do everything that it can do in the AM, but can also replay and fabricate messages using anything it can calculate.

5.1 Authenticators

An MT-authenticator is the key mechanism in the modularisation of the process. It is a transformation that applies to each message flow of a protocol and transforms an AM SK-secure protocol into one in the UM.

Definition 2. *An* authenticator *is a protocol translator that takes an SK-secure protocol in the AM to an SK-secure protocol in the UM. An* MT-authenticator *(message transmission authenticator) is an authenticator that is applied to each separate message sent in the AM.*

Each flow of the protocol in the AM will become a multi-flow sub-protocol in the UM. The resultant secure protocol can be simply a concatenation of all sub-protocols. However such a simple approach generates an inefficient protocol and is of limited interest to us despite its proven security. Fortunately we can collapse flows of the sub-protocols with only one assumption: the proof does not rely on the sequence of the internal flows of a particular MT-authenticator. The process is straightforward and simply combines flows which will be sent in the same direction.

An MT-authenticator is *valid* if running it in the UM has the same effect as a simple message transmission in the AM. What this means is that for any efficient adversary in the UM against the MT-authenticator, there is an efficient adversary in the AM so that the outputs of the two adversaries are computationally indistinguishable. Canetti and Krawczyk [11] have proved the following theorem which is the basis of the methodology. In keeping with the descriptions in this section we again give a less formal version of the result.

Theorem 1 ([11]). *Let π be a secure protocol in the AM. Then the protocol formed by applying a valid MT-authenticator is a secure protocol in the UM.*

We can now summarise the Canetti-Krawczyk approach in the following four steps.

1. Design a basic protocol and prove it is SK-secure in the AM.
2. Design an MT-authenticator and prove that it is a valid authenticator.
3. Apply the MT-authenticator to the basic protocol to produce a protocol that is automatically secure in the UM.
4. As necessary, re-order and re-use message components to optimise the resulting protocol.

5.2 MAC based authenticator

Bellare, Canetti and Krawczyk [3] provided two public key based authenticators, one using signatures and the other using encryption. Since we aim to use an existing shared key (derived non-interactively) for authentication purposes, we will use a simpler MAC based authenticator mentioned by Canetti and Krawczyk [11]. Protocols based on existing shared keys have often seemed less interesting, except with the use of an on-line server. The use of certificated Diffie-Hellman values and identity-based techniques allows us to effectively apply this shared-key authenticator in a public key setting.

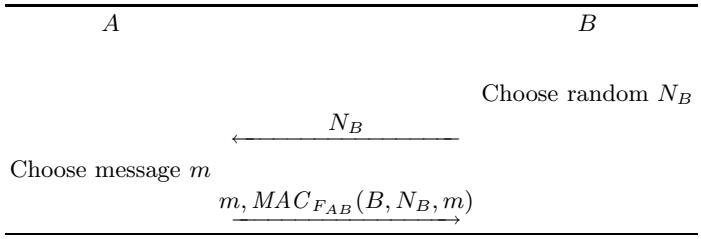


Fig. 1. MAC based authenticator

The MAC based authenticator, shown in Fig. 1, allows A to authenticate a message m which A wishes to send to B. The existing shared key F_{AB} is used as the key for the MAC. It can be proven that, provided F_{AB} is a securely shared key, this is a valid MT-authenticator, which means that when it is applied to the messages of a protocol in the AM, the result is a proven secure protocol in the UM. In the next section we apply this authenticator to two AM protocols to provide protocols with proven security and attractive deniability and efficiency properties.

6 Concrete Key Establishment Protocols

In this section, we sketch concrete key establishment protocols and their security proofs. These are based on the certificated Diffie-Hellman and identity-based approaches to generating shared secrets that we sketched in Section 3. The work of Canetti and Krawczyk actually provides two candidates which we can use to build secure key establishment protocols: the basic Diffie-Hellman protocol and a simple protocol based on public key encryption. We consider each in turn.

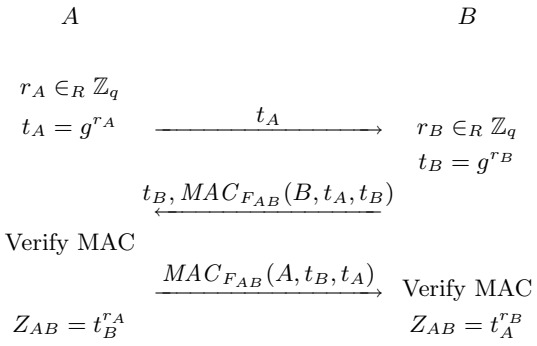
6.1 Key Establishment Using Diffie-Hellman

Protocol 2 is the result of applying a MAC-based authenticator to Diffie-Hellman exchange. In this protocol F_{AB} denotes a shared key that is derived from public information. Because of the use of a (static) shared key, either party can perfectly simulate protocol runs. Thus it enjoys a stronger deniability property than the original SIGMA design, Protocol 1, as we discussed in Section 2.1. The protocol structure is also simpler than Protocol 1. In repeated use of this protocol with the same party, the shared key computation is of course not required. This means that only MAC computations are needed, whereas Protocol 1 requires fresh signature computations for each protocol run. We also note that this protocol is not susceptible to the authentication flaw that SIGMA suffers from. We have three basic possibilities for the shared key F_{AB}:

1. F_{AB} could be obtained from static, certificated Diffie-Hellman values. In order to obtain a bandwidth efficient protocol, both these values and the ephemeral values exchanged in the protocol itself could be defined in an elliptic curve setting. The resulting protocol is bandwidth efficient (though the bits required to exchange certificates should not be discounted).
2. F_{AB} could be obtained using Girault's protocol as described in Section 3.1. Here the advantage is the lack of certificates; however the protocol is not particularly bandwidth-efficient in the light of the observations of [31].
3. F_{AB} could be obtained from the identity-based non-interactive key agreement protocol of Sakai *et al* using a single pairing computation. For this case, the ephemeral Diffie-Hellman session key agreement in Protocol 2 can also use an elliptic curve group. By choosing an appropriate elliptic curve group for the ephemeral values, we can obtain a particularly bandwidth-efficient protocol: with (say) a 128-bit MAC function and an elliptic curve over a 160-bit field with point compression, the total bandwidth for the protocol can be as low as $2 \times (161 + 128) = 578$ bits. This version of the protocol also enjoys our strongest form of deniability, where not even certificates are available to identify possible protocol participants.

By the results of Canetti and Krawczyk, Protocol 2 is secure in their model as long as F_{AB} is a shared secret that is chosen randomly and distributed securely to A and B. Each of the above methods for deriving F_{AB} requires its own assumptions in order to prove security. For example, for the last case the protocol

Shared Information: Fixed key F_{AB} derived from public information.



Protocol 2: Key agreement based on Diffie-Hellman and shared key

of Sakai *et al* was proven secure by Enge and DuPont given that the BDHP is hard. In order to complete the security proof of our protocols we really need to prove that each of the MT-authenticators, with the different choices of the F_{AB} construction, are secure. However, in this paper we make the heuristic assumption that concatenating each of the above F_{AB} constructions with the MT-authentcator of Fig. 1 leads to a new secure MT-authenticator.

6.2 Key Establishment from Public Key Encryption

A second, encryption based, key establishment protocol is also proven secure in the AM by Canetti and Krawczyk. This protocol lacks forward secrecy but can be more efficient than using Diffie-Hellman. The protocol uses only one message in the AM, and basically consists of A choosing a new key and sending it to B by encrypting it under B's public key. Protocol 3 shows the result of applying the MAC based authenticator to the encryption-based AM protocol.

In order for Protocol 3 to be identity-based, the encryption used must also be identity-based. The Boneh-Franklin scheme [6, 7] outlined in Section 4 is ideal for this purpose. (Note that the security proof requires that the encryption algorithm is secure under chosen ciphertext attacks, so the variant in [6] with this property must be used.)

Once again, the shared key F_{AB} used in the MAC can be obtained by one of several methods; the most convenient (given that identity-based encryption is being used) is to make use of the identity-based non-interactive key agreement protocol of Sakai *et al*. Notice that the same public parameters and infrastructure can be used to support both the encryption scheme and the non-interactive key agreement scheme. It is advisable to use different identities for deriving the relevant public and private keys. For example, one might use $H(A||\texttt{encrypt})$ and $H(A||\texttt{share})$ for A's two public keys.

Shared Information: Fixed key F_{AB} derived from identity information. Security parameter t.

A		B
		$N_B \in_R [1..t]$
	$\xleftarrow{N_B}$	
$K \in_R [1..t]$		
	$\xrightarrow{E_B(K), A, N_A, MAC_{F_{AB}}(B, N_B, E_B(K))}$	
		Decrypt to obtain K and verify MAC
	$\xleftarrow{Z_{AB} = MAC_K(A, B, N_A, N_B)}$	
Verify MAC		

Protocol 3: Key transport from identity-based encryption

Given that the protocol of Figure 1 is a valid MT authenticator for this choice of shared key, the security of Protocol 3 follows immediately. (We again make the heuristic assumption that deriving F_{AB} with any of our proposed methods does not break the security proof.) This protocol is also certificateless and perfectly simulatable by both parties, ensuring it has our strongest form of deniability. This protocol does not suffer from the authentication flaw that SIGMA has.

The protocol is also bandwidth and computationally efficient: the computation of F_{AB} and public-key encryption using the scheme of [6] require one pairing each, while the scheme of [6] results in compact encryptions (equal in size to the plaintext plus one element of $\mathbb{G}_1$ and one hash function output). With an appropriately chosen $\mathbb{G}_1$, group elements can be represented with around 260 bits. With a 128 bit session key K, 128 bit nonces N_A, N_B, and 128 bit MACs, the longest message in the protocol has $772 + |A|$ bits and the total communication is just 1028 bits.

7 Conclusion

We have described two identity-based protocols whose security is based on the generic security proofs of Canetti and Krawczyk. Although both these protocols arise naturally by applying the identity-based primitives to the building blocks of the Canetti-Krawczyk method, neither of them seems to have been published before. They have attractive properties that are not possessed by other protocols that have been considered in the literature. In particular, they have strong deniability properties and can be computationally and bandwidth efficient.

In future work we aim to complete the formal security proof of our protocols by proving that the MT-authenticator formed by first deriving F_{AB}, and then

applying the MAC-based authenticator, remains valid. We also believe that it is worthwhile to explore more deeply the optimisation of these novel protocols with specific concrete algorithms.

Acknowledgements

We would like to thank Neil Dunbar of Hewlett-Packard Company for helpful discussions on IPSec and IKE.

References

1. P. S. L. M. Barreto, H.Y. Kim, B. Lynn, and M. Scott. Efficient algorithms for pairing-based cryptosystems. In *Advances in Cryptology - CRYPTO 2002*, LNCS. Springer-Verlag, 2002.
2. P.S.L.M. Barreto, B. Lynn, and M. Scott. Constructing elliptic curves with prescribed embedding degrees. In *Security in communication networks – SCN'2002*, volume 2576 of *LNCS*, pages 263–273. Springer-Verlag, 2002.
3. M. Bellare, R. Canetti, and H. Krawczyk. A modular approach to the design and analysis of authentication and key exchange protocols. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 419–428. ACM Press, 1998. Full version at `http://www-cse.ucsd.edu/users/mihir/papers/key-distribution.html`.
4. M. Bellare and P. Rogaway. Entity authentication and key distribution. In *Advances in Cryptology – CRYPTO'93*, pages 232–249. Springer-Verlag, 1993. Full version at `www-cse.ucsd.edu/users/mihir`.
5. M. Bellare and P. Rogaway. Provably secure session key distribution – the three party case. In *Proceedings of the 27th ACM Symposium on the Theory of Computing*, 1995.
6. D. Boneh and M. Franklin. Identity-based encryption from the Weil pairing. In J. Kilian, editor, *Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference*, volume 2139 of *LNCS*, pages 213–229. Springer Verlag, 2001.
7. D. Boneh and M. Franklin. Identity-based encryption from the Weil pairing. *SIAM J. Computing*, 32(3):586–615, 2003. `http://www.crypto.stanford.edu/~dabo/abstracts/ibe.html`, full version of [6].
8. D. Boneh, H. Shacham, and B. Lynn. Short signatures from the Weil pairing. In C. Boyd, editor, *Advances in Cryptology – ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 514–532. Springer-Verlag, 2001.
9. M.S. Borella. Methods and protocols for secure key negotiation using IKE. *IEEE Network*, pages 18–29, July/August 2000.
10. M. Burrows, M. Abadi, and R. Needham. A logic of authentication. *Proceedings of the Royal Society*, A426:233–271, 1989.
11. R. Canetti and H. Krawczyk. Analysis of key-exchange protocols and their use for building secure channels. In *Advances in Cryptology – Eurocrypt 2001*, volume 2045 of *LNCS*, pages 453–474. Springer-Verlag, 2001. `http://eprint.iacr.org/2001/040.pg.gz`.
12. R. Canetti and H. Krawczyk. Security analysis of IKE's signature-based key-exchange protocol. In *Advances in Cryptology – Crypto 2002*, 2002.

13. P.-C. Cheng. An architecture for the Internet Key Exchange protocol. *IBM Systems Journal*, 40(3):721–745, 2001.
14. W. Diffie and M. Hellman. New directions in cryptography. *IEEE Transaction on Information Theory*, 22:644–654, 1976.
15. R. Dupont and A. Enge. Practical non-interactive key distribution based on pairings. Cryptology ePrint Archive, Report 2002/136, 2002. `http://eprint.iacr.org/`.
16. R. Dupont, A. Enge, and F. Morain. Building curves with arbitrary small MOV degree over finite prime fields. Cryptology ePrint Archive, Report 2002/094, 2002. `http://eprint.iacr.org/`.
17. N. Ferguson and B. Schneier. A cryptographic evaluation of IPsec. `http://www.counterpane.com/ipsec.html`, 2000.
18. S. D. Galbraith. Supersingular curves in cryptography. In C. Boyd, editor, *Proceedings of AsiaCrypt 2001, 7th International Conference on the Theory and Application of Cryptology and Information Security*, volume 2248 of *LNCS*, pages 495–513. Springer-Verlag, 2001.
19. S. D. Galbraith, K. Harrison, and D. Soldera. Implementing the Tate pairing. In *Algorithmic Number Theory 5th International Symposium, ANTS-V*, volume 2369 of *LNCS*, pages 324–337. Springer-Verlag, 2002.
20. M. Girault. Self-certified public keys. In *Advances in Cryptology - Eurocrypt 1991*, LNCS, pages 490–497. Springer-Verlag, 1991.
21. D. Harkins and D. Carrel. *The Internet Key Exchange (IKE)*, November 1998. Internet RFC 2409.
22. D. Harkins, C. Kaufman, T. Kivinen, S. Kent, and R. Perlman. *Design Rationale for IKEv2*, February 2002. Internet Draft.
23. A. Joux. A one round protocol for tripartite Diffie-Hellman. In W. Bosma, editor, *Algorithmic Number Theory, IV-th Symposium (ANTS IV), Lecture Notes in Computer Science 1838*, pages 385–394. Springer-Verlag, 2000.
24. H. Krawczyk. SIGMA: The SIGn and MAc approach to authenticated Diffie-Hellman and its use in the IKE protocols. `http://www.ee.technion.ac.il/~hugo/sigma.html`.
25. G. Lowe. Some new attacks upon security protocols. In *9th IEEE Computer Security Foundations Workshop*, pages 162–169. IEEE Computer Society Press, 1996.
26. W. Mao and K.G. Paterson. On the plausible deniability feature of Internet protocols. `http://www.isg.rhul.ac.uk/~kp/IKE.ps`, 2002.
27. A.J. Menezes, T. Okamoto, and S.A. Vanstone. Reducing elliptic curve logarithms to a finite field. *IEEE Trans. Info. Theory*, 39:1636–1646, 1983.
28. R.M. Needham and M.D. Schroeder. Using encryption for authentication in large networks of computers. *Communications of the ACM*, 21(12):993–999, December 1978.
29. K.G. Paterson. Cryptography from pairings: A snapshot of current research. *Information Security Technical Report*, 7(3):41–54, 2002. `http://www.isg.rhul.ac.uk/~kp/pairings.ps`.
30. R. Perlman and C. Kaufman. Key exchange in IPSec: Analysis of IKE. *IEEE Internet Computing*, pages 50–56, November-December 2000.
31. S. Saeednia. A note on Girault's self-certified model. *Information Processing Letters*, 86:323–327, 2003.
32. R. Sakai, K. Ohgishi, and M. Kasahara. Cryptosystems based on pairing. In *The 2000 Sympoium on Cryptography and Information Security*, Okinawa, Japan, January 2000.

33. A. Shamir. Identity-based cryptosystems and signature schemes. In *Proceedings of Crypto 84*, pages 47–53. Springer-Verlag, 1985.
34. E. R. Verheul. Evidence that XTR is more secure than supersingular elliptic curve cryptosystems. In B. Pfitzmann, editor, *Advances in Cryptology — Proceedings of EUROCRYPT'01, Lecture Notes in Computer Science 2045*, pages 195–210. Springer-Verlag, 2001.
35. J. Zhou. Fixing a security flaw in IKE protocols. *Electronics Letters*, 35(13):1072–1073, 24th June 1999.
36. J. Zhou. Further analysis of the Internet key exchange protocol. *Computer Communications*, 23:1606–1612, 2000.

Appendix

Unfortunately, the basic SIGMA protocol and its derivatives suffers from an authentication flaw similar to that developed by Lowe in [25]. In the attack, Mallory masquerades as other principals. Mallory masquerades as B to A and persuades A to initiate a protocol with him. Mallory communicates with B as himself in an incomplete protocol run.

1. $A \rightarrow$ Mallory("B"): g^{r_A}
   $1'$ Mallory $\rightarrow B$: g^{r_A}
   $2'$ $B \rightarrow$ Mallory: g^{r_B}, B, $Sig_B(g^{r_B}, g^{r_A})$, $MAC_{K_m}(B)$
2. Mallory("B") $\rightarrow A$: g^{r_B}, B, $Sig_B(g^{r_B}, g^{r_A})$, $MAC_{K_m}(B)$
3. $A \rightarrow$ Mallory("B"): A, $SIG_A(g^{r_A}, g^{r_B})$, $MAC_{K_m}(A)$
   $3'$ dropped

At the end of the attack, I believes he has successfully completed a protocol run with R; in fact his run was with Mallory. On the other hand, R believes he has engaged in an incomplete protocol run with Mallory. The flaw we have demonstrated is *not* just a simple case of Mallory stopping the last message in a protocol run between A and B – that attack is possible against any protocol and is not particularly interesting.

For further discussion about this flaw and its consequences for building Denial of Service attacks on SIGMA and related protocols, we refer the reader to [26].