ID-based Signatures from Pairings on Elliptic Curves

Kenneth G. Paterson

Information Security Group,
Mathematics Department, Royal Holloway University of London,
Egham, Surrey TW20 0EX, UK.
Kenny.Paterson@rhul.ac.uk

Abstract. We present an efficient identity-based signature scheme which makes use of bilinear pairings on elliptic curves. Our scheme is similar to the generalized ElGamal signature scheme. We consider the security of our scheme.

Keywords: Identity-based cryptography, signatures, Weil pairing, Tate pairing.

1 Introduction

The concept of identity-based cryptography is due to Shamir [1]. Recently, Boneh and Franklin presented an identity-based encryption scheme based on properties of the Weil and Tate pairings on elliptic curves [2, 3]. Their scheme appears to be the first fully functioning, efficient and provably secure identity-based encryption scheme. Such a scheme has the property that a user's public-key is an easily calculated function of his identity, while a user's private key can be calculated for him by a trusted authority. For reasons of efficiency and convenience, it is desirable to have an identity-based signature scheme (where the signature verification function is easily obtained from identity) which is able to make use of the same underlying computational primitives and possibly the same keys. Such a scheme, predating [2], was presented in [4]. Here we present an identity-based signature scheme that is more computationally efficient than the scheme of [4] and consider its security properties. Our scheme is similar to the generalized ElGamal signature scheme [5, Section 11.73]. We note that our scheme is quite distinct from the ordinary (i.e. non-identity-based) signature scheme considered in [6].

2 Notation

We use the same notation as in [3]. We let G_1 be an additive group of prime order q and G_2 be a multiplicative group of the same order q. We assume the existence of a bi-linear map $\hat{e}$ from $G_1 \times G_1$ to G_2 with the property that the discrete logarithm problems in both G_1 and G_2 are hard (in a sense made precise

in [3]). Typically, G_1 will be a subgroup of the group of points on an elliptic curve over a finite field, G_2 will be a subgroup of the multiplicative group of a related finite field and the map $\hat{e}$ will be derived from the Weil or Tate pairing on the elliptic curve. We also assume that an element $P \in G_1$ satisfying $\hat{e}(P, P) \neq 1_{G_2}$ is known. We refer to [3, 7] for a fuller description of how these groups, maps and other parameters should be selected in practice for efficiency and security.

We let ID be a string denoting the identity of a user and H_1, H_2 and H_3 be public cryptographic hash functions. We require $H_1 : \{0,1\}^* \to G_1$, $H_2 : \{0,1\}^* \to \mathbb{Z}_q$ and $H_3 : G_1 \to \mathbb{Z}_q$. In our scheme, a user's public key for signature verification is $Q_{ID} = H_1(ID)$, while his secret key for signature generation is $D_{ID} = s \cdot Q_{ID}$, where $s \in \mathbb{Z}_q$ is a system-wide master secret known to a trusted authority. These keys are the same as in the encryption scheme of [3]. If desired, encryption and signature keys can be separated simply by concatenating the string ID with extra bits which identify the keys' intended functions. We also assume that the value $P_{pub} = s \cdot P$ is publicly known.

3 The Scheme

To sign a message M (in the form of a bit-string of arbitrary length), a user first chooses a random $k \in \mathbb{Z}_q^*$ and computes his signature on message M as the pair $(R, S) \in G_1 \times G_1$, where:

$$R = k \cdot P, \quad S = k^{-1}(H_2(M) \cdot P + H_3(R) \cdot D_{ID}).$$

Here k^{-1} is the inverse of k in $\mathbb{Z}_q^*$.

To verify a purported signature (U, V) on message M, the verifier computes $\hat{e}(U, V)$ and compares it to the value $\hat{e}(P, P)^{H_2(M)} \cdot \hat{e}(P_{pub}, Q_{ID})^{H_3(R)}$. The signature is accepted if these values in G_2 match, and rejected otherwise.

Notice that if (R, S) is a valid signature on M, then we have

$$\begin{aligned} \hat{e}(R, S) &= \hat{e}(k \cdot P, k^{-1}(H_2(M) \cdot P + H_3(R) \cdot D_{ID})) \\ &= \hat{e}(P, H_2(M) \cdot P + H_3(R) \cdot D_{ID}) \\ &= \hat{e}(P, P)^{H_2(M)} \cdot \hat{e}(P_{pub}, Q_{ID})^{H_3(R)}. \end{aligned}$$

where we have used the bi-linearity properties of $\hat{e}$. Thus a valid signature will always satisfy the check.

Our scheme is similar to the generalized ElGamal signature scheme [5, Section 11.73]. In that scheme, R is a group element and S is defined via arithmetic in $\mathbb{Z}_q$ while verification takes place over a group of order q. By contrast, in our scheme R and S are both defined over a group (typically, a subgroup of the group of points on an elliptic curve) and verification is carried out via the map $\hat{e}$ over a second group. The inclusion of $H_3(R)$ is necessary in the generalised ElGamal scheme to prevent some standard attacks. These attacks do not appear to apply to our scheme, and it is possible to omit the term $H_3(R)$ from our scheme without any apparent loss of security. However, this adaptation has the property that if (R, S) is a valid signature on M, then so too is $(\ell \cdot R, \ell^{-1} \cdot S)$ for any $\ell \in \mathbb{Z}_q^*$. This 'homomorphic' property does not appear to help an attacker compute signatures on new messages.

4 Efficiency

To compute a signature requires only two hash-function evaluations, some computation in G_1, and an inversion modulo q. Signature generation does not require computation of the map $\hat{e}$.

The cost of verifying a signature is dominated by computations of the pairing $\hat{e}$. Notice that $\hat{e}(P, P)$ is a fixed value in G_2 that can be computed once and stored. Thus one $\hat{e}$ computation can be saved. Notice too that the value of $\hat{e}(P_{pub}, Q_{ID})$ does not depend on the particular message M and so is fixed when verifying any particular user's signatures. Therefore the cost of computing this pairing can be amortised over many verifications of that user's signatures. In this situation, we can justifiably claim that our scheme requires only a single $\hat{e}$ computation. To verify a signature also requires two hash-function evaluations, two exponentiations in G_2 and one multiplication in G_2.

Two or three pairing computations are required to verify a signature in the scheme of [4] (depending on whether or not many signatures are being verified for a fixed signer). Thus our scheme requires one pairing computation less than the scheme of [4].

Our signatures have size twice the size of group elements in G_1. Standard point compression techniques can be used to reduce their size by a factor of 2.

5 Security

The standard model for studying the security of signature schemes is that of [8]. There an adversary $\mathcal{A}$ is challenged with a fixed public key, is allowed to adaptively request signatures on messages of his choice and is tasked to produce an existential forgery for that key, i.e. a valid signature for any previously unrequested message. To capture security in the identity-based setting, we extend this model by additionally allowing $\mathcal{A}$ to obtain private keys D_{ID} corresponding to identities ID of his choice and to request signatures on messages and for identities of his choice. The adversary's task is now to produce a signature on a message and identity of his choice, but not for an identity for which he has requested the private key, and not for a message/identity combination for which he has already requested a signature. The adversary's *advantage* is the probability that his final output is accepted as a valid signature for his choice of message and identity.

We consider the security of our scheme against such an extended adversary in the random oracle model [9]. Suppose then that the hash function H_1 is replaced by a random function in our scheme. Then we can show, using techniques similar to those in the proof of [3, Lemma 4.6], that an adversary $\mathcal{A}$ with advantage ϵ against our scheme can be used to build an adversary $\mathcal{B}$ who can produce forgeries (in the sense of [8]) for a related, non-identity-based signature scheme with advantage ϵ/cN_1. Here N_1 is the number of H_1 queries made by $\mathcal{A}$ and c is a small constant. In this related scheme, the fixed public key is Q_{ID}, the corresponding private key is D_{ID} and our verification condition holds. This

ordinary signature scheme resembles the generalized ElGamal signature scheme [5, Section 11.73]. Thus the security of our identity-based scheme is linked to the security of an ordinary signature scheme which resembles a well-known scheme. If the ordinary scheme is secure, then so is ours, and we can say that the ability to make private key extractions is of essentially no use to an adversary $\mathcal{A}$.

6 Conclusion

We have presented an identity-based signature scheme using the same computational primitives (and keys if so desired) as the identity-based encryption scheme of Boneh and Franklin [3]. Our scheme is more efficient than a previous scheme [4] and we have related the security of our scheme to that of a non identity-based signature scheme that closely resembles the generalized ElGamal signature scheme.

Acknowledgment

I am grateful to Simon Blackburn for comments which helped to improve the presentation of the paper.

References

1. A. Shamir, "Identity-based cryptosystems and signature schemes" in *Proc. Crypto'84*, LNCS Vol. 196, Springer, pp.47-53, 1985.
2. D. Boneh and M. Franklin, "Identity-Based Encryption from the Weil pairing," in *Proc. Crypto 2001*, LNCS Vol. 2139, Springer, pp. 213-229, 2001.
3. D. Boneh and M. Franklin, "Identity-Based Encryption from the Weil pairing," full version, available from `http://crypto.stanford.edu/ dabo/abstracts/ibe.html`
4. R. Sakai, K. Ohgishi and M. Kasahara, "Cryptosystems based on pairing," 2000 Symposium on Cryptography and Information Security (SCIS2000), Okinawa, Japan, Jan. 26-28, 2000.
5. A.J. Menezes, P.C. van Oorschot and S.A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1996.
6. D. Boneh, M. Franklin and H. Shacham, "Short signatures from the Weil pairing," in *Proc. AsiaCrypt 2001*, LNCS Vol. 2248, Springer, pp. 514-532, 2001.
7. S.D. Galbraith, "Supersingular curves in cryptography," in *Proc. AsiaCrypt 2001*, LNCS Vol. 2248, Springer, pp.495-513, 2001.
8. S. Goldwasser, S. Micali and R. Rivest, "A digital signature scheme secure against adaptive chosen-message attacks," *SIAM J. Computing*, Vol. 17(2), pp. 281-308, April 1988.
9. M. Bellare and P. Rogaway, "Random oracles are practical: a paradigm for designing efficient protocols," in *Proc. First ACM Conference on Computer and Communications Security*, pp. 62-73, 1993.