ID-based signatures

Alexey Vyskubov

1 Introduction

In 1984 Shamir [1] introduced identity-based encryption scheme (IBE). Identity-based systems are public-key encryption schemes, but users do not have explicit public keys. Instead, public key is derived, in a publicly known way, from a user's publicly available identity (for example, name or e-mail address).

Quite natural idea is to use the same philosophy for signing: verifier should be able easy retrieve signature-verification function from signer identity.

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 computational primitives and possibly the same keys as encryption scheme. In [2] some ID-based signature scheme close to the recent Boneh and Franklin encryption scheme [3] was proposed, but no security arguments were given.

In this survey we review articles [4, 5].

The first article presents on identity-based signature scheme that is more computationally efficient than the scheme of [2] and also discuss some security properties.

The second article proposes an identity-based signature scheme using gap Diffie-Hellman groups. Using GDH groups obtained from bilinear pairings it is possible to obtain an ID-based signature scheme that shares the same system parameters and the same private/public key pairs with the ID-based encryption scheme [3]. The security is based on hardness of computational Diffie-Hellman problem. More precisely, under the random oracle model, the scheme is proved to be secure against existential forgery on adaptively chosen message and ID attack, which is a natural ID-based version of the standard adaptively chosen message attack, assuming CDHP is intractable.

2 ID-based signatures from pairings on elliptic curves

2.1 Notation

We will use the following objects:

- G_1 is an additive group of prime order q.
- G_2 is a multiplicative group of the same order q.
- $\hat{e} : G_1 \times G_1 \to G_2$ is a bilinear pairing.
- $P \in G_1$ is a known element such that $\hat{e}(P, P) \neq 1_{G_2}$.

We suppose that discrete logarithm problem is hard in G_1 and in G_2.

Typically, G_1 is a subgroup of the group of points on an elliptic curve over a finite field, G_2 is a subgroup of the multiplicative group of a related finite field and the map $\hat{e}$ is derived from the Weil or Tate pairing on the elliptic curve.

2.2 ID-based signing phases

ID-based signing requires trusted third party and consists of four phases.

1. Setup

   Trusted third party creates public system parameters and secret master key.

2. Extract

   Given an identity, trusted third party computes private key for given identity using master key and gives this private key to the identity owner.

3. Signing

   Using system parameters and private key, received from TTP, signer creates message signature.

4. Verify

   Verifier computes public key for the given identity using system parameters and checks the signature using it.

2.3 Setup

TTP chooses G_1, P, G_2. The master key is random $s \in Z_q$. System parameters are $(G_1, G_2, P, P_{pub} = sP, \hat{e}, H_1, H_2, H_3)$ where H_i are cryptographic hash functions:

$$H_1 : \{0,1\}^* \to G_1,$$

$$H_2 : \{0,1\}^* \to Z_q,$$

$$H_3 : G_1 \to Z_q.$$

2.4 Extract

The private key for the given identity ID is

$$D_{ID} = sH_1(ID).$$

2.5 Sign

To sign a message $M \in \{0,1\}^*$ a user first chooses a random $k \in Z_q^*$ and computes his signature on message M as the pair $(kP, k^{-1}S) \in G_1 \times G_1$ where

$$S = H_2(M)P + H_3(kP)D_{ID}.$$

Here k^{-1} is the inverse of k in Z_q^*.

2.6 Verify

To verify a purported signature (U, V) on message M the verifier first computes the public key for the given identity ID:

$$Q_{ID} = H_1(ID).$$

Then he computes $\hat{e}(U, V)$ and compares the result to the value $\hat{e}(P, P)^{H_2(M)}\hat{e}(P_{pub}, Q_{ID})^{H_3(U)}$.

2.7 Sign: correctness

If (U, V) is a valid signature, then

$$\hat{e}(U, V) = \hat{e}(kP, k^{-1}S) = \hat{e}(P, S) =$$
$$= \hat{e}(P, aP + bD_{ID}) =$$
$$= \hat{e}(P, aP) \cdot \hat{e}(P, bD_{ID}) =$$
$$= \hat{e}(P, P)^a \cdot \hat{e}(P, D_{ID})^b =$$
$$= \hat{e}(P, P)^{H_2(M)} \cdot \hat{e}(P, sQ_{ID})^b =$$
$$= \hat{e}(P, P)^{H_2(M)} \cdot \hat{e}(sP, Q_{ID})^{H_3(kP)} =$$
$$= \hat{e}(P, P)^{H_2(M)} \cdot \hat{e}(P_{pub}, Q_{ID})^{H_3(kP)}$$
$$= \hat{e}(P, P)^{H_2(M)} \cdot \hat{e}(P_{pub}, Q_{ID})^{H_3(U)}.$$

Thus a valid signature will always satisfy the check.

We should also notice that if (U, V) is a valid signature for M then $lU, l^{-1}V$ is a valid signature for M as well, but this "homomorphic" property does not appear to help an attacker compute signatures on new messages.

2.8 Efficiency

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

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 thus 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 computating this pairing can be amortised over many verifications of that user's signatures. In this situation, we can justifiable 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 [2].

2.9 Security

We will not discuss the security of described scheme in this survey. The author claims (but shows no exact proof) that the security of this scheme is linked to the security of an ordinary signature scheme which resembles a well-known scheme.

3 An identity-based signature from gap Diffie-Hellman groups

In this section we present ID-based signature scheme which can be built on the base of any gap group.

3.1 Setup

Master key is random $s \in Z_q^*$.

System parameters are $(G, P, P_{pub}, H_1, H_2)$ where G is a gap group of prime order q, P is a generator of G, $P_{pub} = sP$, H_i are cryptographic hash functions:

$$H_1 : \{0,1\}^* \times G \to Z_q,$$

$$H_2 : \{0,1\}^* \to G.$$

Notice that in a security proof H_i are considered as random oracles.

3.2 Extract

Given an identity ID, TTP computes the private key:

$$D_{ID} = sH_2(ID).$$

3.3 Sign

To sign a message $M \in \{0,1\}^*$ a user first chooses a random $k \in Z_q^*$ and computes his signature on message M as the pair (U, V) where

$$U = kH_2(ID),$$

$$V = (k + H_1(M, U))D_{ID}.$$

3.4 Verify

To verify a purported signature (U, V) on message M the verifier first computes the public key for the given identity ID:

$$Q_{ID} = H_2(ID).$$

Then he checks whether

$$(P, P_{pub}, U + H_1(M, U)Q_{ID}, V)$$

is a valid Diffie-Hellman tuple.

3.5 Sign: correctness

Let $h = H_1(M, U)$. If (U, V) is a valid signature then

$$U = kQ_{ID}$$

$$V = (k + h)D_{ID} = s(k + h)Q_{ID}$$

$$(P, P_{pub}, U + H_1(M, U)Q_{ID}, V) =$$

$$= (P, sP, (k + h)Q_{ID}, s(k + h)Q_{ID})$$

3.6 Attack model

The most general known notion of security of a non-ID-based signature scheme is security against existential forgery on adaptively chosen message attacks; in this model, an adversary wins the game if he outputs a valid pair of message and a signature, where he is allowed to ask the signer to sign any message except the output. We consider the following natural generalisation of this notion, which is acceptable as a standard model of security for ID-based signature schemes. We say that an ID-based signature scheme, which consists of four algorithms Setup, Extract, Sign and Verify playing the same role as ours, is *secure against existential forgery an adaptively chosen message and ID attacks* if no polynomial time algorithm $\mathcal{A}$ has a non-negligible advantage against a challenger $\mathcal{C}$ in the following game:

1. $\mathcal{C}$ runs Setup of the scheme. The resulting system parameters are given to $\mathcal{A}$.
2. $\mathcal{A}$ issues the following queries as he wants:

(a) Hash function query. $\mathcal{C}$ computes the value of the hash function for the requested input and sends the value to $\mathcal{A}$.

(b) Extract query. Given an identity ID, $\mathcal{C}$ returns the private key corresponding to ID which is obtained by running Extract.

(c) Sign query. Given an identity ID and a message M, $\mathcal{C}$ returns a signature which is obtained by running Sign.

3. $\mathcal{A}$ outputs (ID, M, σ) where ID is an identity, M is a message, and σ is a signature, such that ID and (ID, M) are not equal to the inputs of any query to Extract and Sign, respectively. $\mathcal{A}$ wins the game if σ is a valid signature of M for ID.

3.7 Reduction to given ID attack

Consider the following variant of the above game: First we fix an identity ID. In Step 1, $\mathcal{C}$ gives to $\mathcal{A}$ system parameters together with ID, and in Step 3, $\mathcal{A}$ must output the given ID (together with a message and a signature) as its final result. If no polynomial time algorithm $\mathcal{A}$ has non-negligible advantage in this game, we say that the signature scheme is secure under existential forgery on adaptively chosen message and given ID attack. The following lemma shows that the problem can be reduced to this case.

Lemma 1

If there is an algorithm $\mathcal{A}_0$ for an adaptively chosen message and ID attack to our scheme with running time t_0 and advantage ε_0, then there is an algorithm $\mathcal{A}_1$ for an adaptively chosen message and given ID attack which has running time $t_1 \leq t_0$ and advantage $\varepsilon_1 \geq \varepsilon_0(1 - 1/q)/q_{H_2}$, where q_{H_2} is the maximum number of queries to H_2 asked by $\mathcal{A}_0$. In addition, the numbers of queries to hash functions, Extract, and Sign asked by $\mathcal{A}_1$ are the same as those for $\mathcal{A}_0$.

Idea of proof

The idea of proof is quite simple. Let us create some other challenger for $\mathcal{A}_0$ which will work exactly as the original challenger except the r-th query of $\mathcal{A}_0$ to H_2 (r is chosen at random in $\{1, \ldots, q_{H_2}\}$). Instead of returning $H_2(\text{asked ID})$ this new challenger will return $H_2(ID)$ as a result of r-th query. (ID is the identity given for $\mathcal{A}_1$). Then if $\mathcal{A}_0$ returns the identity asked in r-th step we check if the message and the signature together with given ID is a valid forged signature, return this signature. If it is not, or $\mathcal{A}_0$ outputs the different identity, return fail.

Now few simple observations prove the lemma:

1. The probability that the final output is valid (if we did not output fail) is at least ε, because $\mathcal{A}_0$ learns nothing from query results (H_2 is a random oracle).

2. Since H_2 is a random oracle, the probability that $\mathcal{A}_0$ asked for hash of identity it outputs (in the case if $\mathcal{A}_0$ output is a valid forged signature) is at least $1 - 1/q$.

3. Since r is independently and randomly chosen, the probability that $\mathcal{A}_0$ returns the identity for which he made r-th hash query is at least $1/q_{H_2}$.

   Combining these, we find that the probability that built algorithm will return the given identity together with a message and valid signature is at least

$$\frac{\varepsilon_0 \left(1 - \frac{1}{q}\right)}{q_{H_2}},$$

as desired.

3.8 Reduction to CDHP

The next step reduces the given ID attack to computational Diffie-Hellman problem.

Lemma 2

If there is an algorithm $\mathcal{A}_1$ for an adaptively chosen message and given ID attack to our scheme which queries H_1, H_2, Sign and Extract at most q_{H_1}, q_{H_2}, q_S and q_E times, respectively, and has running time t_1 and advantage $\varepsilon_1 \geq 10(q_S+1)(q_S+q_{H_1})/q$, then CDHP can be solved with probability $\varepsilon_2 \geq 1/9$ within running time $t_2 \leq 23 q_{H_1} t_1/\varepsilon_1$.

As it is quite logical to consider $\log q$ as security parameter, the lemma 2 combined with lemma 1 mean that an adversary cannot have non-negligible advantage against the original ID-based signature scheme if CDHP is hard in G.

We will not provide the proof of lemma 2 in this survey as it is quite complex and refers quite often to technical details of proofs in [6].

References

[1] A. Shamir. Identity-based cryptosystems and signature schemes. Lecture Notes in Computer Science, Vol. 1966, Springer-Verlag, pp. 537-554, 1999

[2] R. Sakai, K. Ohgishi and M. Kasahara. Cryptosystems based on pairing. 2000 Symposium on Cryptography and Information Security, Okinawa, Japan, Jan. 26-28, 2000.

[3] D. Boneh and M. Franklin. Identity-Based Encryption from the Weil pairing. `http://crypto.stanford.edu/ dabo/pubs.html`

[4] Kenneth G. Peterson. ID-based Signatures from Pairings on Elliptic Curves.

[5] Jae Choon Cha and Jung Hee Cheon. An Identity-Based Signature from Gap Diffie-Hellman Groups.

[6] D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. J. of Cryptology, Vol. 13, pp. 361-396, 2000.