A Secure Signature Scheme from Bilinear Maps

Dan Boneh[⋆] Ilya Mironov[⋆⋆] Victor Shoup

Computer Science Department
Stanford University
{dabo,mironov}@cs.stanford.edu

Courant Institute
New York University
shoup@cs.nyu.edu

Abstract. We present a new class of signature schemes based on properties of certain bilinear algebraic maps. These signatures are secure against existential forgery under a chosen message attack in the standard model (without using the random oracle model). Security is based on the computational Diffie-Hellman problem. The concrete schemes that we get are the most efficient provable discrete-log type signature schemes to date.

1 Introduction

Provably secure signature schemes can be constructed from the most basic cryptographic primitive, one-way functions [NY89,Rom90]. As is often the case with cryptographic schemes designed from elementary blocks, this signature scheme is somewhat impractical. Over the years several signature schemes were proposed based on stronger complexity assumptions. The most efficient schemes provably secure in the standard model are based on the Strong RSA assumption [GHR99,CS99].

Surprisingly, no scheme based on any discrete logarithm problem comes close to the efficiency of the RSA-based schemes. We give a partial solution to this open problem using bilinear maps. A bilinear map is a function $e\colon \mathbb{G}_0 \times \mathbb{G}_1 \mapsto \mathbb{G}_2$ that is consistent with group operations in both of its arguments, as described in the next section. Our construction gives an existentially unforgeable signature whenever the Computational Diffie-Hellman (CDH) assumption holds in $\mathbb{G}_0 \times \mathbb{G}_1$, that is, no efficient algorithm can compute $g^\alpha \in \mathbb{G}_1$ given the three values $h, h^\alpha \in \mathbb{G}_0$, and $g \in \mathbb{G}_1$. Precise definitions are given in the next section.

Our signature scheme is based on a signature authentication tree with a large branching factor. At a high level, our construction bares some resemblance to the signature schemes of Dwork-Naor [DN94] and Cramer-Damgård [CD96]. Both these schemes are based on the hardness of factoring whereas our scheme is based on CDH.

We obtain a concrete signature scheme by instantiating the bilinear map with the modified Weil or Tate pairings. This is the only known provably secure signature scheme based on the CDH assumption that is more efficient than

⋆ Supported by NSF Career Award and Packard Foundation.
⋆⋆ Supported by Microsoft Fellowship.

the most general constructions of [CD95]. It is interesting to note that recently, Lysyanskaya [Lys02] constructed a verifiable unpredictable function (VUF) secure under the Generalized Diffie-Hellman assumption in the standard model. Such functions (defined in [MRV99]) provide a special type of secure signatures called unique signatures. This construction gives an excellent VUF, but as a signature scheme it compares poorly with the construction of [CD95]. We review other known signature schemes in Section 4.

Bilinear maps such as the Weil or Tate pairing have recently been used to construct a number of new cryptosystems, including three-party key exchange [Jou00], identity based encryption [BF01], short signatures [BLS01], credential systems [Ver01], hierarchical identity based encryption [HL02,GS02], and others. In this paper we show how bilinear maps can be used to construct efficient signature schemes secure in the standard model.

Efficient discrete log based signature schemes are known to exist in the random oracle model [PS96,BLS01]. Security in the random oracle model does not imply security in the real world. In this paper we only study signature schemes secure in the standard complexity model.

2 Mappings with Algebraic Properties

We consider binary maps between groups that are consistent with the group structure of their arguments. Such binary maps are called bilinear. Their formal definition follow.

Definition 1 (Bilinear map). *A function* $e\colon \mathbb{G}_0 \times \mathbb{G}_1 \mapsto \mathbb{G}_2$ *is* bilinear *if for any four elements* $g_1, g_2 \in \mathbb{G}_0$*,* $H_1, H_2 \in \mathbb{G}_1$ *the following holds:*

$$e(g_1 \circ g_2, H_1) = e(g_1, H_1) \circ e(g_2, H_1) \quad \textit{and} \quad e(g_1, H_1 \circ H_2) = e(g_1, H_1) \circ e(g_1, H_2).$$

In this paper we intentionally limit our scope to finite cyclic groups, which allows us to give more efficient constructions.

Throughout the paper we use the following notation. Small Roman letters $f, g, h, \ldots$ from the lower part of the alphabet denote elements of the group $\mathbb{G}_0$; capital Roman letters $F, G, H, \ldots$ stand for elements of $\mathbb{G}_1$; elements of $\mathbb{G}_2$ are denoted by letters from the end of the alphabet x, y, z.

Our constructions are based on the Computational Diffie Problem (CDH) defined below. However to simplify the exposition we define the notion of a secure bilinear map. We then show that this notion is equivalent to CDH. Informally, we say that a bilinear map $e\colon \mathbb{G}_0 \times \mathbb{G}_1 \to \mathbb{G}_2$ is secure if, given $g \in \mathbb{G}_0$ and $G, H \in \mathbb{G}_1$, it is hard to find $h \in \mathbb{G}_0$ such that $e(h, H) = e(g, G)$. More precisely, we define secure bilinear maps as follows.

Definition 2 (Secure bilinear map). *A bilinear map* $e\colon \mathbb{G}_0 \times \mathbb{G}_1 \mapsto \mathbb{G}_2$ *is* (t, ε)-secure *if for all* t*-time adversaries* $\mathcal{A}$

$$\mathsf{AdvBLM}_{\mathcal{A}} = \Pr\left[e\Big(\mathcal{A}(g, G, H),\ H\Big) = e(g, G) \;\middle|\; g \xleftarrow{R} \mathbb{G}_0; G, H \xleftarrow{R} \mathbb{G}_1 \right] < \varepsilon.$$

The probability is taken over the coin tosses of the algorithm $\mathcal{A}$ and random choice of g, G, H.

We given two simple examples of bilinear maps that are believed to be secure.

- $e(\cdot,\cdot)\colon \mathbb{G}_0 \times \mathbb{G}_1 \mapsto \mathbb{F}^*_{p^r}$ is the Weil or Tate pairings on an elliptic curve $E/\mathbb{F}_p$ and $\mathbb{G}_0, \mathbb{G}_1$ are distinct subgroups of $E[q]$ for some prime q (recall that $E[q]$ is the subgroup containing all point of order dividing q on some elliptic curve $E/\mathbb{F}_p$). For certain curves $E/\mathbb{F}_p$ one can slightly modify the Weil pairing (as in [BF01]) so that we may take $\mathbb{G}_0 = \mathbb{G}_1$. At any rate, the security of the bilinear map $e(\cdot,\cdot)$ is equivalent to the Computational Diffie-Hellman assumption (CDH) in $\mathbb{G}_0 \times \mathbb{G}_1$. Informally, the CDH assumption in $\mathbb{G}_0 \times \mathbb{G}_1$ means that:

  It is infeasible to find G^a given random group elements $h, h^a \in \mathbb{G}_0$, and $G \in \mathbb{G}_1$. When $\mathbb{G}_0 = \mathbb{G}_1$ this is the standard CDH assumption in $\mathbb{G}_0$.
- Another bilinear map believed to be secure is $r(\cdot,\cdot)\colon \mathbb{Z}^*_N \times \mathbb{Z}^+_{\varphi(N)} \mapsto \mathbb{Z}^*_N$ defined as $r(g, H) = g^H$, where N is a product of two primes. This map is secure under the Strong RSA assumption [BP97]. Briefly, the Strong RSA assumption says that the following problem is difficult to solve:

  For a random $x \in Z^*_N$ find $r > 1$ and $y \in Z^*_N$ such that $y^r = x$.

  We give a short argument why the security of the map $r(\cdot,\cdot)$ is reducible to the Strong RSA assumption. Suppose the map $r(\cdot,\cdot)$ is insecure. Then, given (G, H, g) it is feasible to find an $h \in \mathbb{G}_0$ such that $r(g, G) = r(h, H)$, i.e. $g^G = h^H$. This solution yields (through application of the Extended Euclidean algorithm) a $z \in \mathbb{Z}^*_N$ satisfying $z^a = g$, where $a = H/\gcd(G, H)$. For random $G, H \xleftarrow{R} \mathbb{Z}^+_{\varphi(N)}$ the probability that $a = 1$ is negligible. Therefore breaking the security of the bilinear map r amounts to breaking the Strong RSA assumption. It is not known whether the converse is true.

Next, we show that for finite order groups a bilinear map $e\colon \mathbb{G}_0 \times \mathbb{G}_1 \to \mathbb{G}_2$ is secure if and only if the CDH assumption holds in $\mathbb{G}_0 \times G_1$. We first precisely define the CDH assumption.

Definition 3 (Computational Diffie-Hellman problem). *The* Computational Diffie-Hellman problem *is (t, ε)-hard if for all t-time adversaries $\mathcal{A}$ we have*

$$\mathsf{AdvCDH}_{\mathcal{A}} = \Pr\left[\mathcal{A}(g, H, H^a) = g^a \;\middle|\; g \xleftarrow{R} \mathbb{G}_0; H \xleftarrow{R} \mathbb{G}_1; a \xleftarrow{R} \mathbb{Z}_{|\mathbb{G}_1|}\right] < \varepsilon.$$

Claim. Suppose that $\mathbb{G}_0, \mathbb{G}_1, \mathbb{G}_2$ are cyclic groups of prime order p. Suppose the map $e\colon \mathbb{G}_0 \times \mathbb{G}_1 \to \mathbb{G}_2$ is non-degenerate in the following sense: $e(h, H) \neq 1$ for some $h \in \mathbb{G}_0$ and $H \in \mathbb{G}_1$. Then the map e is (t, ε)-secure if and only if the CDH problem is (t, ε)-hard.

Proof. First, suppose CDH can be solved in time t with probability at least ε. We give an algorithm to show that the map is not (t, ε)-secure. Let $g \in \mathbb{G}_0$ and $G, H \in \mathbb{G}_1$ where both $G, H \neq 1$. We wish to find $h \in \mathbb{G}_0$ such that

$e(g, G) = e(h, H)$. Since $\mathbb{G}_1$ is cyclic of prime order p there exists an $a \in \mathbb{Z}_p$ such that $G = H^a$. Let $h = g^a$. Then h satisfies

$$e(h, H) = e(g^a, H) = e(g, H^a) = e(g, G).$$

Therefore, $h = g^a$, which is the solution to the CDH problem (g, H, G), is the required h. Hence, if the map is (t, ε)-secure then CDH is (t, ε)-hard.

Conversely, suppose there is a t-time algorithm that given random (g, G, H) outputs $h \in \mathbb{G}_0$ such that $e(g, G) = e(h, H)$ with probability at least ε. We show how to solve CDH. Let (g, H, G) be a random instance of the CDH problem, where $H \neq 1$. Write $G = H^a$ for some $a \in \mathbb{Z}_p$. Let h be such that $e(g, G) = e(h, H)$. Then

$$e(h, H) = e(g, G) = e(g, H^a) = e(g^a, H)$$

and hence $e(h/g^a, H) = 1$. Since $H \neq 1$ it follows that $h = g^a$, since otherwise the map e would be degenerate. Hence, if CDH is (t, ε)-hard then the map is (t, ε)-secure. $\square$

3 Security for signature schemes

We recall the standard definition of secure signature schemes stated in terms of exact security, in the spirit of [BR94]. This notion of existential unforgeability under adaptive chosen-message attack is due to [GMR88].

A signature scheme is a triple of probabilistic algorithms: a key generation algorithm KeyGen, signer $\mathsf{Sign}(SK, Msg)$, and verifier $\mathsf{Verify}(Msg, Sig, PK)$. By convention, Verify outputs 1 if it accepts the signature and 0 otherwise. We use the oracle notation, where $A^{B(\cdot)}(\cdot)$ means A supplied with oracle access to B.

Definition 4. *A signature scheme is* (t, ε, n)-existentially unforgeable under adaptive chosen-message attack*, if for all pairs of algorithms* $\mathcal{F}_1, \mathcal{F}_2$ *running in time at most* t

$$\mathsf{AdvSig}_{\mathcal{F}_1,\mathcal{F}_2} = \Pr[\mathsf{Verify}(\mathcal{F}_2(T), PK) = 1 \mid (SK, PK) \leftarrow \mathsf{KeyGen}(1^k);\ T \leftarrow \mathcal{F}_1^{\mathsf{Sign}(SK,\cdot)}(1^k)] < \varepsilon.$$

$\mathcal{F}_1$ *requests no more than* n *signatures from* Sign *and the message output by* $\mathcal{F}_2$ *is different from the messages signed by* Sign*. The probability is taken over the coin tosses of* KeyGen*,* Sign*,* $\mathcal{F}_1$ *and* $\mathcal{F}_2$*. Here* $\mathcal{F}_2(T)$ *outputs a message/signature pair.*

4 Previous work

The seminal paper [GMR84] formulated a strong notion of security for signature schemes: existential unforgeability under adaptive chosen-message attacks. Since then there have been many proposals for signature schemes meeting this notion of security based on different assumptions and of varying efficiency. Some of these results are summarized in Table 1. With an exception of GMR, all schemes have running time of the signing and verification algorithms proportional to the signature length. This information is omitted from the table.

Reference	**Signature length**	**Public key length**	**Max number of signatures**	**Security assumption**
[GMR84]	$O(k\ell)$	$O(k)$	2^ℓ	claw-free trapdoor permutations
[NY89]	$O(k^2\ell)$	$O(k)$	2^ℓ	UOWHF
[CD95]	$O(k\ell)$	$O(k)$	2^ℓ	one-way homomorphism
[DN94]	$O(k\ell)$	$O(kn)$	n^ℓ	RSA assumption
[CD96]	$O(k\ell)$	$O(k+n)$	n^ℓ	RSA assumption
[GHR99],[CS99]	$O(k)$	$O(k)$	∞	Strong RSA assumption
[Lys02]	$O(km)$	$O(km)$	2^m	Generalized Diffie-Hellman
this paper	$O(k\ell)$	$O(kn)$	n^ℓ	Computational Diffie-Hellman (secure bilinear maps)

Table 1. Summary of provably secure signature schemes. k is the security parameter, ℓ is the depth of the authentication three, n is the branching factor of the tree, and m is the message length. The O-notation refers to asymptotics as a function of the security parameter k.

A signature scheme may be based on the most general cryptographic assumption, namely that one-way functions exist [NY89,Rom90]. The proof proceeds via constructing an authentication tree and results in a scheme in which the signature length is proportional to the binary logarithm of the total number of messages signed with the same public key. More efficient (in terms of the signature length) signature schemes can be based on the Strong RSA assumption or its variants [CS99,GHR99]. Important steps in constructing these schemes were the Dwork-Naor scheme [DN94] later improved by [CD96]. Both schemes use trees with a large branching factor, which are therefore very shallow.

The Dwork-Naor trick is crucial for understanding this paper. In a nutshell, the trick allows to increase the tree's branching factor without blowing up the signature size. An authentication-tree scheme produces signatures that represent paths connecting messages and the root of the tree. Messages are usually placed in the very bottom level of the tree, though [CD95] puts messages in the leaves hanging from the inner nodes. The authentication mechanism works inductively: the root authenticates its children, they authenticate their children, and so on, down to the message authenticated by its parent. If the authentication mechanism allows attaching children to a node only when some secret information is known, then the adversary cannot forge signatures without knowing that secret.

[GMR84] and [CD95] take similar approaches in designing the authentication mechanism (hence the identical asymptotic of their signature length). They concatenate bit representations of a node's children and compute a value that authenticates this string in respect to the node. To verify a node's authenticity we must know all siblings of the node. If the tree is binary, the signature contains twice as many nodes as the the depth of the tree. Indeed, each node must be accompanied by its sibling.

Since the signature length is proportional to the depth of the tree, one may wonder whether increasing the tree's branching factor is a good idea. The following simple computation shows why this is counterproductive.

Suppose one wants to sign N messages. If the authentication tree is binary, its depth must be at least $\log_2 N$. The signature length is roughly $2k \log_2 N$, where k is the security parameter that accounts for the nodes' size. When the branching factor of the tree is increased from 2 to d, the depth of the tree goes down to $\log_d N$. The signatures, however, must include all siblings of the nodes on the authentication path (ancestors of the message-leaf). Thus the signature size becomes $dk \log_d N$, which is actually *more* than in the binary case.

The improvement achieved by [DN94] is due to the way the sibling nodes are authenticated. Using a stronger complexity assumption than in the GMR scheme, authenticity of a node can be verified given its parent and its authentication value, whose size is independent of the branching factor. Each node has the authentication value different from those of its siblings and their authenticity can be verified independently. It allows to increase the number of a node's children and decrease the depth of the tree without inflating the signature length. The public key size, being the branching factor *times* the security parameter and, because of that, the main drawback of [DN94], was reduced in the Cramer-Damgård scheme [CD96]. Finally, two independent methods proposed in [CS99] and [GHR99] make the tree flat by letting the signer (but not the adversary) add branches on the fly.

We do not distinguish between statefull and stateless schemes. All schemes can be made memoryless at the price of doubling the tree depth [Gol86]. To do so [Gol86] suggested using a pseudo-random function (PRF) to generate all secret values in internal nodes of the authentication tree. The key for the PRF is kept secret at the signer. Furthermore, instead of sequentially stepping through the leaves of the tree (one leaf per signature) we pick a random leaf for every signature. To make sure the same leaf is not chosen at random for two different signatures we need to square the number of leaves and hence double the depth of the tree.

In this paper we implement the Dwork-Naor method using a secure bilinear map. Improving the scheme further in the direction of [CD96,CS99,GHR99] is left as an open problem.

5 The new signature scheme

We present a signature scheme based on a secure bilinear map. An instantiation of this construction yields the most efficient provably secure scheme based on the Computational Diffie-Hellman assumption known to date.

The signature scheme is designed as follows.

- **Setup of the scheme.** Groups $\mathbb{G}_0$ and $\mathbb{G}_1$ have prime order p. Bilinear map $e\colon \mathbb{G}_0 \times \mathbb{G}_1 \mapsto \mathbb{G}_2$ is secure. $\mathcal{H}_k\colon \mathcal{M} \mapsto \{0,1\}^s$ is a family of collision-resistant hash functions, where $\mathcal{M}$ is the message space (the key for $\mathcal{H}_k$ is fixed at random during key generation and made public). The signature scheme allows signing of ℓ^n messages, where ℓ and n are arbitrary positive integers.
- **Key generation.** The public and private keys are created as follows:
  Step 1. Pick $\alpha_1, \ldots, \alpha_n \overset{R}{\leftarrow} \mathbb{Z}_p^*$ and $H \overset{R}{\leftarrow} \mathbb{G}_1$. Choose a random key k for the collision resistant hash function $\mathcal{H}_k$. Compute $H_1 \leftarrow H^{1/\alpha_1}, \ldots, H_n \leftarrow H^{1/\alpha_n} \in \mathbb{G}_1$.
  Step 2. Pick $g \overset{R}{\leftarrow} \mathbb{G}_0$. Compute $y \leftarrow e(g, H)$.
  Step 3. Pick $\beta_0 \overset{R}{\leftarrow} \mathbb{Z}_p$. Compute $x_0 \leftarrow e(g, H)^{\beta_0}$.
  Step 4. The **public key** is $k, H, H_1, \ldots, H_n, y$ and x_0. The **private key** is $\alpha_1, \ldots, \alpha_n, \beta_0$, and g.
- **Signing algorithm.** Each node in the tree is authenticated with respect to its parent; messages to be signed are authenticated with respect to the leaves, which are selected in sequential order and never reused. To sign the i^{th} message $M \in \mathcal{M}$ the signer generates the i^{th} leaf of the authentication tree together with a path from the leaf to the root. We denote the leaf by $x_\ell \in \mathbb{G}_1$ and denote the path from the leaf to the root by $(x_\ell, i_\ell, x_{\ell-1}, i_{\ell-1}, \ldots, i_1, x_0)$, where x_j is the i_j^{th} child of x_{j-1} (here $i_j \in \{1, \ldots, n\}$). The leaf and the nodes along the path and their authentication values are computed as follows:
  Step 1. All nodes of the tree, including the leaf x_ℓ, that have not been visited before are computed as $x_j \leftarrow e(g, H)^{\beta_j}$ for some random $\beta_j \overset{R}{\leftarrow} \mathbb{Z}_p$. The secret β_j is stored for as long as node x_j is an ancestor of the current signing leaf (β_ℓ is discarded immediately after use). Note that when using stateless signatures [Gol86] the β_j are derived using a PRF and hence there is no need to remember their values.
  Step 2. The authentication value of x_j, the i_j^{th} child of x_{j-1}, is $f_j \leftarrow g^{\alpha_{i_j}(\beta_{j-1} + \mathcal{H}_k(x_j))}$.
  Step 3. The authentication value of $\mathcal{H}_k(M)$, the child of x_ℓ, is $f \leftarrow g^{\beta_\ell + \mathcal{H}_k(M)}$.
  Step 4. The **signature** on M is $(f, f_\ell, i_\ell, \ldots, f_1, i_1)$.
- **Verification algorithm.** To verify a signature $(\hat{f}, \hat{f}_\ell, \hat{i}_\ell, \ldots, \hat{f}_1, \hat{i}_1)$ on a message M we reconstruct the nodes $\hat{x}_\ell, \ldots, \hat{x}_0$ in a bottom-up order (from leaf $\hat{x}_\ell$ to root $\hat{x}_0$). The signature is accepted if and only if $\hat{x}_0$ matches the root of the tree. More precisely, to verify the signature the verifier performs the following steps:
  Step 1. Compute $\hat{x}_\ell \leftarrow e(\hat{f}, H) \cdot y^{-\mathcal{H}_k(M)}$.
  Step 2. For $j = \ell \ldots 1$ compute $\hat{x}_{j-1} \leftarrow e(\hat{f}_j, H_{i_j}) \cdot y^{-\mathcal{H}_k(\hat{x}_j)}$.
  Step 3. The signature is accepted if $\hat{x}_0 = x_0$.

A signature output by the signing algorithm passes the verification test. Indeed, step 1 of the verification algorithm results in

$$e(f, H) \cdot y^{-\mathcal{H}_k(M)} = e(g^{\beta_\ell + \mathcal{H}_k(M)}, H) \cdot e(g, H)^{-\mathcal{H}_k(M)} = e(g^{\beta_\ell}, H) = x_\ell.$$

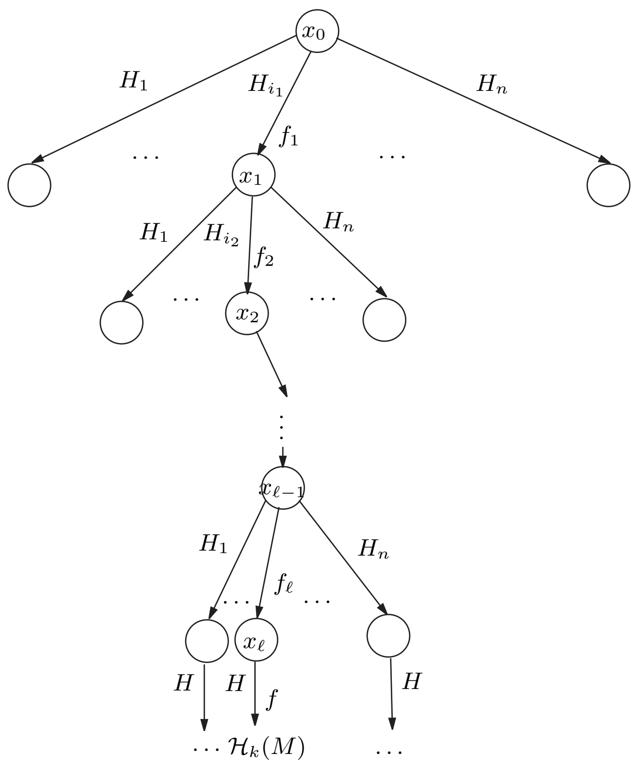

Fig. 1. Authentication tree. The signature on M is $(f, f_\ell, i_\ell, \ldots, f_1, i_1)$.

For any $j \in 1 \ldots \ell$ the result of computation in step 2 of the verification algorithm is

$$e(f_j, H_{i_j}) \cdot y^{-\mathcal{H}_k(x_j)} = e(g^{\alpha_{i_j}(\beta_{j-1}+\mathcal{H}_k(x_j))}, H^{1/\alpha_{i_j}}) \cdot e(g, H)^{-\mathcal{H}_k(x_j)} =$$
$$e(g^{\beta_{j-1}+\mathcal{H}_k(x_j)}, H) \cdot e(g^{-\mathcal{H}_k(x_j)}, H) = e(g^{\beta_{j-1}}, H) = x_{j-1}.$$

Signature length. Suppose the user needs to generate a billion signatures. Then taking $n = 20$ and $\ell = 4$ is sufficient ($4^{20} > 10^{12}$). The public key will contain 20 elements in $\mathbb{G}_1$ and two elements in $\mathbb{G}_2$. The signature contains five elements in $\mathbb{G}_0$. To get stateless signatures we would need to double the depth so that each signature will contain nine elements of $\mathbb{G}_0$.

Security. We show that the signature scheme is secure against the most general attack. To formalize it as a claim we need to recall the definition of collision-resistance and assume the existence of a collision-finding algorithm for e.

Definition 5 (Collision-resistant function). *We call function $\mathcal{H}\colon \mathcal{K}\times\mathcal{M}\mapsto \{0,1\}^s$ a family of* (t,ε)-collision-resistant *functions, if for any t-time algorithm $\mathcal{A}$ its advantage in finding a collision*

$$\mathsf{AdvCR}_{\mathcal{A}} = \Pr[\mathcal{H}_k(M_1) = \mathcal{H}_k(M_2), M_1 \neq M_2 \mid k \stackrel{R}{\leftarrow} \mathcal{K}; (M_1, M_2) \leftarrow \mathcal{A}(k)] < \varepsilon.$$

The probability is taken over $\mathcal{A}$'s coin tosses.

Definition 6 (Collision-finding algorithm). *We say that an algorithm is* collision-finding *for bilinear map $e\colon \mathbb{G}_0\times\mathbb{G}_1 \mapsto \mathbb{G}_2$ if it outputs a solution $g', g'' \in \mathbb{G}_0$, where $g', g'' \neq 1$, to the equation*

$$e(g', G') = e(g'', G''),$$

for given $G', G'' \in \mathbb{G}_1$ whenever such a solution exists.

For the bilinear maps defined on elliptic curves, namely the Weil and Tate pairings, it is easy to build collision finding algorithms. This yields a concrete discrete-log signature scheme as described later in this section.

Theorem 1. *The signature scheme is $(t, \varepsilon/(n+1), m)$-secure against existential forgery against adaptive chosen-message attack under the following assumptions:*

- *e is a (t,ε)-secure bilinear map;*
- *$\mathcal{H}$ is (t,ε)-collision-resistant function;*
- *there is an efficient collision-finding algorithm for e.*

Proof. The proof is by contradiction. We show that existence of an efficient forger implies that we can either find a collision in $\mathcal{H}$ or solve

$$e(g^*, G_1) = e(g_2, G_3)$$

for g^* given $G_1, G_3 \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_0$.

To this end we set up the public key to help the simulator in answering the adversary's signing queries. In the same time a forgery would let us respond to the challenge or find a collision of the hash function with a non-negligible probability.

First, we set $y \leftarrow e(g_2, G_3)$. Second, we pick random $i \in \{0,\dots,n\}$. Consider two cases.

Case 1: $i = 0$**.** We assign $H \leftarrow G_1$, pick $\gamma_j \stackrel{R}{\leftarrow} \mathbb{Z}_p$ for all $j \in 1\dots n$ and assign $H_j \leftarrow G_3^{1/\gamma_i}$. All internal nodes of the authentication tree, including the root x_0 but excluding the leaves, are computed as random powers of $e(g_2, G_3)$. Suppose x' is the j^{th} child of x'', where $x'' = e(g_2, G_3)^\gamma$. The authentication value for x' is computed as $f' \leftarrow g_2^{\gamma_j(\gamma+\mathcal{H}(x'))}$. It correctly authenticates x' as the j^{th} child of x'', since

$$e(f', H_j)\cdot y^{-\mathcal{H}(x')} = e(g_2^{\gamma_j(\gamma+\mathcal{H}(x'))}, G_3^{1/\gamma_j})\cdot e(g_2, G_3)^{-\mathcal{H}(x')} = e(g_2, G_3)^\gamma = x''.$$

When the adversary requests a signature on a message M, the path (i_ℓ, $x_{\ell-1}$, $i_{\ell-1}$, …, i_1,x_0) is generated and the authentication values for all internal nodes along the path are included in the signature. The leaf x_ℓ is computed as $x_\ell \leftarrow e(g_2, G_1^\gamma \cdot G_3^{-\mathcal{H}(M)})$ for a randomly chosen $\gamma \stackrel{R}{\leftarrow} \mathbb{Z}_p$. The authentication value for $\mathcal{H}(M)$ is set to be $f \leftarrow g_2^\gamma$. The leaf x_ℓ is authenticated as the i_ℓ^{th} child of $x_{\ell-1}$ as above. This results in a valid signature $(f, f_\ell, i_\ell, \ldots, f_1, i_1)$.

Case 2: $i \in \{1, \ldots, n\}$**.** Assign $H_i \leftarrow G_1$. We randomly choose $\gamma, \gamma_1, \ldots,$ $\gamma_{i-1}, \gamma_{i+1}, \ldots, \gamma_n \stackrel{R}{\leftarrow} \mathbb{Z}_p$, assign $H \leftarrow G_3^{1/\gamma}$ and $H_j \leftarrow G_3^{1/\gamma_j}$, for all $j \neq i$. We apply the collision-finding algorithm that returns d_1 and d_2 satisfying

$$e(d_1, G_1) = e(d_2, G_3).$$

The authentication tree is constructed from the bottom up. Suppose we are given a set of siblings $z_1, \ldots, z_n$. The challenge is to define their parent node z such that we may find authentication values for all of them. Let $z \leftarrow e(d_2, G_3^\delta) y^{-\mathcal{H}(z_i)}$ for a randomly chosen $\delta \stackrel{R}{\leftarrow} \mathbb{Z}_q$. For all $j \in 1 \ldots n$, such that $j \neq i$, the authentication value of z_j can be computed as $f_j \leftarrow (g_2^{\mathcal{H}(z_j)-\mathcal{H}(z_i)} \cdot d_1^\delta)^{\gamma_j}$. Indeed, for all such j

$$\begin{aligned} e(f_j, H_j) \cdot y^{-\mathcal{H}(z_j)} = e((g_2^{\mathcal{H}(z_j)-\mathcal{H}(z_i)} \cdot d_2^\delta)^{\gamma_j}, G_3^{1/\gamma_j}) \cdot e(g_2, G_3)^{-\mathcal{H}(z_j)} = \\ e(g_2^{\mathcal{H}(z_j)-\mathcal{H}(z_i)} \cdot d_2^\delta, G_3) \cdot e(g_2^{-\mathcal{H}(z_j)}, G_3) = e(g_2^{-\mathcal{H}(z_j)} \cdot d_2^\delta, G_3) = \\ e(d_2, G_3^\delta) \cdot e(g_2, G_3)^{-\mathcal{H}(z_i)} = e(d_2, G_3^\delta) y^{-\mathcal{H}(z_i)} = z_j. \end{aligned}$$

Node z_i is authenticated with $f_i \leftarrow d_1^\delta$. Indeed,

$$e(f_i, H_i) \cdot y^{-\mathcal{H}(z_i)} = e(d_1^\delta, G_1) \cdot y^{-\mathcal{H}(z_i)} = e(d_2^\delta, G_3) \cdot y^{-\mathcal{H}(z_i)} = z.$$

It follows that every internal node may be computed given its j^{th} child. In particular, the root of the tree, x_0, is determined by values on the path (x_ℓ, j, $x_{\ell-1}$,…,x_1, j), which can be efficiently computed by the randomized algorithm given above.

When a signature is requested by the adversary, a new leaf is generated as $x_\ell \leftarrow e(g_2, H^\delta)$, where $\delta \stackrel{R}{\leftarrow} \mathbb{Z}_p$. Then the path to the root is computed, which would involve generating new nodes from their j^{th} children. The authentication value for $\mathcal{H}(M)$ is given by $f \leftarrow g_2^{\delta+\gamma\mathcal{H}(M)}$.

We have shown that the simulator may effectively replace the signing oracle and answer the adversary's queries. The simulated answers have the same distribution as signatures output by the real signer.

Solving the challenge. Let us consider an algorithm that interacts with the signing oracle and then forges a signature $(f, f_\ell, i_\ell, \ldots, f_1, i_1)$ on a message M. Without loss of generality we may assume that the adversary makes ℓ^n queries. Denote the full authentication tree constructed by the simulator by T. The signature on M has the form of an authenticated path up the tree from a leaf to the root x_0.

Let $(x_\ell, i_\ell, x_{\ell-1}, i_{\ell-1}, \ldots, i_1, x_0)$ be the path reconstructed from the signature. Define x_j as the lowest node of the path that also appears in T.

We distinguish between two types of forgeries:

Type I: $j = \ell$. Denote the message that was previously authenticated using x_ℓ by M' and let f' be the authentication value of $\mathcal{H}(M')$. It follows that

$$e(f, H) \cdot y^{-\mathcal{H}(M)} = x_\ell,$$
$$e(f', H) \cdot y^{-\mathcal{H}(M')} = x_\ell,$$

from which we have

$$y^{\mathcal{H}(M) - \mathcal{H}(M')} = e(f/f', H). \tag{1}$$

Type II: $j < \ell$. Let the x'_{j+1} be the i_j^{th} child of x_j in T and f'_j be its authentication value. Similarly, there are two equations

$$e(f_j, H_{i_j}) \cdot y^{-\mathcal{H}(x_{j+1})} = x_j,$$
$$e(f'_j, H_{i_j}) \cdot y^{-\mathcal{H}(x'_{j+1})} = x_j,$$

that imply

$$y^{\mathcal{H}(x_{j+1}) - \mathcal{H}(x'_{j+1})} = e(f_j/f'_j, H_{i_j}). \tag{2}$$

Consider two possibilities. If $\mathcal{H}(M) = \mathcal{H}(M')$ in type I or $\mathcal{H}(x_{j+1}) = \mathcal{H}(x'_{j+1})$ in type II forgery, then we find a collision in the hash function. Otherwise, we solved equation (1) or (2) of the type

$$y^d = e(\hat{f}, \hat{H}),$$

for d and $\hat{f}$, where $d \neq 0$ and $\hat{H}$ is one of $H, H_1, \ldots, H_n$.

Recall that $y = e(g_2, G_3)$. Since the simulation of the adversary's view is perfect, the probability that $\hat{H} = G_1$ is $\frac{1}{n+1}$. Thus $e(\hat{f}^{1/d}, G_1) = e(g_2, G_3)$ and $\hat{f}^{1/d} = g^*$, i.e. a solution to the challenge.

Therefore an efficient algorithm for forging a signature can be used to either find a collision in the hash function or disprove security of e. This completes the proof of the theorem. $\square$

6 Concrete signature scheme

To obtain a concrete secure signature scheme we instantiate the bilinear map $e\colon \mathbb{G}_0 \times \mathbb{G}_1 \to \mathbb{G}_2$ with a map for which there is an efficient collision finding algorithm. Let $E/\mathbb{F}_p$ be an elliptic curve. We denote by $E(\mathbb{F}_{p^r})$ the group of points of E over $\mathbb{F}_{p^r}$. Let q be a prime divisor of $|E(\mathbb{F}_p)|$.

When using a supersingular curve over $\mathbb{F}_p, p > 3$ we can take $\mathbb{G}_0 = \mathbb{G}_1$ as the subgroup containing all points of order q in $E(\mathbb{F}_p)$. The group $\mathbb{G}_2$ is the subgroup of order q of $\mathbb{F}^*_{p^2}$. The modified Weil pairing [BF01], denoted $\hat{e}$, is a non-degenerate bilinear map on $\mathbb{G}_0 \times \mathbb{G}_1$. Since the CDH assumption is believed to hold for the group $\mathbb{G}_0$, we know by Claim 2 that $\hat{e}$ is secure in the sense of Definition 2. Since the modified Weil pairing is symmetric (i.e. $\hat{e}(G,H) = \hat{e}(H,G)$ for all $H,G \in \mathbb{G}_0$) a collision finding algorithm for $\hat{e}$ is immediate: given $G,H \in \mathbb{G}_1$ we output $H,G \in \mathbb{G}_0$ as a collision. Indeed, $\hat{e}(G,H) = \hat{e}(H,G)$ and hence H,G is a collision for G,H. Since $\hat{e}$ is secure and there is an efficient collision finder we obtain the following corollary to Theorem 1:

Corollary 1. *The signature scheme of Section 5 instantiated with the modified Weil pairing $\hat{e}$ is existentially unforgeable under a chosen message attack if the CDH assumption holds for the group $\mathbb{G}_0 = \mathbb{G}_1$ defined above.*

To obtain shorter signatures one can avoid using supersingular curves and instead use a family of curves due to Miyaji et al. [MNT01]. Curves $E/\mathbb{F}_p, p > 3$ in this family are not supersingular and have the property that if q divides $|E(\mathbb{F}_p)|$ then $E[q]$ is contained in $E(\mathbb{F}_{p^6})$. Let $\mathbb{G}_0$ be the subgroup of order q in $E(\mathbb{F}_p)$ and let $\mathbb{G}_1$ be some other subgroup of order q in $E(\mathbb{F}_{p^6})$. The Weil pairing e on $\mathbb{G}_0 \times \mathbb{G}_1$ is not degenerate. Furthermore, since CDH is believed to be hard on $\mathbb{G}_0 \times \mathbb{G}_1$, we know by Claim 2 that e is a secure bilinear map in the sense of Definition 2. To build a collision finder for e we use the trace map $\mathrm{tr}\colon E(\mathbb{F}_{p^6}) \to E(\mathbb{F}_p)$. For almost all choices of $\mathbb{G}_1$ the map tr defines an isomorphism from $\mathbb{G}_1$ to $\mathbb{G}_0$. Then, a collision finder for e works as follows: given $(G,H) \in \mathbb{G}_1$ it outputs as a collision the pair $(\mathrm{tr}(G), \mathrm{tr}(H)) \in \mathbb{G}_0$. Indeed, one can verify that $e(\mathrm{tr}(G), H) = e(\mathrm{tr}(H), G)$. Therefore, by Theorem 1 our signature scheme is secure when using this family of curves.

We note that the RSA function $r(x,d) = x^d \mod N$ while being bilinear does not satisfy the condition of Theorem 1 since the orders of groups $\mathbb{G}_0$ and $\mathbb{G}_1$ are not known to the simulator. Nevertheless, the signature scheme can be instantiated with this function, which would yield a scheme similar to the Dwork-Naor scheme.

7 Conclusion

We presented a new signature scheme secure in the standard model (i.e. without random oracles) using groups in which the Computation Diffie-Hellman assumption holds. Our scheme can be implemented using any secure bilinear map (secure in the sense of Definition 2). Instantiating our signature scheme using the Weil or Tate pairings gives the most efficient known discrete-log type signature secure without random oracles.

References

[BP97] N. Barić and B. Pfitzmann, "Collision-free accumulators and fail-stop signature schemes without trees," Proc. of Eurocrypt'97, pp. 480-494, 1997.

[BF01] D. Boneh and M. Franklin, "Identity based encryption from the Weil pairing," Proc. of CRYPTO'01, pp. 213-229, 2001. Also `http://eprint.iacr.org/2001/090/`

[BLS01] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the Weil pairing," Proc. of Asiacrypt'01, pp. 514-532, 2001.
[BR94] M. Bellare and P. Rogaway, "Optimal asymmetric encryption—how to encrypt with RSA," Proc. of Eurocrypt'94, pp. 92–111, 1994.
[CD95] R. Cramer and I. Damgård, "Secure signature schemes based on interactive protocols," Proc. of CRYPTO'95, pp. 297–310, 1995.
[CD96] R. Cramer and I. Damgård, "New generation of secure and practical RSA-based signatures," Proc. of CRYPTO'96, pp. 173–185, 1996.
[CS99] R. Cramer and V. Shoup, "Signature schemes based on the Strong RSA assumption," Proc. of ACM CCS'99, pp. 46–51, 1999. Full version appeared in ACM TISSEC, v. 3(3), pp. 161–185, 2000.
[DN94] C. Dwork and M. Naor, "An efficient existentially unforgeable signature scheme and its applications," Proc. of CRYPTO'94, pp. 234–246, 1994. Full version appeared in J. of Cryptology, v. 11(2), pp. 187–208, 1998.
[FFS88] U. Feige, A. Fiat, and A. Shamir, "Zero-knowledge proofs of identity," J. of Cryptology, v. 1, pp. 77-94, 1988.
[GHR99] R. Gennaro, S. Halevi, and T. Rabin, "Secure hash-and-sign signatures without the random oracle," Proc. of Eurocrypt'99, pp. 123–139, 1999.
[GS02] C. Gentry and A. Silverberg, "Hierarchical ID-based cryptography", Proc. of Asiacrypt'02, pp. 548–566, 2002.
[Gol86] O. Goldreich, "Two remarks concerning the Goldwasser-Micali-Rivest signature scheme," Proc. of CRYPTO'86, pp. 104–110, 1986.
[GMR84] S. Goldwasser, S. Micali, and R. Rivest, "A 'paradoxical' solution to the signature problem (extended abstract)," Proc. of FOCS'84, pp. 441–448, 1984. Journal version in [GMR88].
[GMR88] S. Goldwasser, S. Micali, and R. Rivest, "A digital signature scheme secure against adaptive chosen-message attacks," SIAM J. on Computing, 17(2), pp. 281–308, 1988.
[HL02] J. Horwitz and B. Lynn, "Towards hierarchical identity-based encryption", Proc. of Eurocrypt'02, pp. 466–481, 2002.
[Jou00] A. Joux, "A one-round protocol for tripartite Diffie-Hellman," Proc. of ANTS'00, pp. 385–394, 2000.
[Lys02] A. Lysyanskaya, "Unique signatures and verifiable random functions from DH-DDH separation," Proc. of CRYPTO'02, pp. 597–612, 2002.
[MRV99] S. Micali, M. Rabin, and S. Vadhan, "Verifiable random functions," Proc. of FOCS'99, pp. 120–130, 1999.
[MNT01] A. Miyaji, M. Nakabayashi, and S. Takano, "New explicit condition of elliptic curve trace for FR-reduction," IEICE Trans. Fundamentals, v. E84 A(5), May 2001.
[NY89] M. Naor and M. Yung, "Universal one-way hash functions and their cryptographic applications," Proc. of STOC'89, pp. 33–43, 1989.
[PS96] D. Pointcheval and J. Stern, "Security proofs for signature schemes," in Proc. of Eurocrypt'96, pp. 387–398, 1996.
[Rom90] J. Rompel, "One-way functions are necessary and sufficient for secure signatures," Proc. of STOC'90, pp. 387–394, 1990.
[Ver01] E. Verheul, "Self-Blindable Credential Certificates from the Weil Pairing," Proc. of Asiacrypt'01, pp. 533–551, 2001.