Contemporary Mathematics

Applications of Multilinear Forms to Cryptography

Dan Boneh and Alice Silverberg

This paper is dedicated to the memory of Ruth I. Michler

Abstract. We study the problem of finding efficiently computable non-degenerate multilinear maps from G_1^n to G_2, where G_1 and G_2 are groups of the same prime order, and where computing discrete logarithms in G_1 is hard. We present several applications to cryptography, explore directions for building such maps, and give some reasons to believe that finding examples with $n > 2$ may be difficult.

1. Introduction

This paper studies some questions in linear algebra and cryptography. Interesting problems in cryptography have recently been solved using Weil or Tate pairings on supersingular elliptic curves, or more generally on supersingular abelian varieties [**27**]. These applications include one-round three-party key exchange [**15**], identity-based encryption [**3**], and short digital signatures [**4**] (see also [**28, 23**]).

We show that multilinear generalizations of Weil or Tate pairings would have far-reaching consequences in cryptography. Section 3 describes the desired properties for a multilinear form. Sections 4 to 6 give several applications. Such forms would enable secure broadcast encryption with very short broadcasts and private keys, a unique signature scheme, and one-round multi-party key exchange. The main question is how to build the required multilinear maps. We now have the means and the opportunity. But do we have the motive? In Section 7 we explore the question of whether multilinear generalizations of Weil or Tate pairings can come from geometry, or even just from a "motive", in the sense of [**14**]. We give evidence that it might not be possible to find cryptographically useful multilinear forms within the realm of algebraic geometry (i.e., coming from an underlying curve, surface, or higher-dimensional variety), except for the case of bilinear pairings on

2000 *Mathematics Subject Classification.* Primary 94A60, Secondary 14G50.

Key words and phrases. pairing-based cryptography, elliptic curve cryptography, abelian variety cryptography.

Boneh thanks the Packard Foundation and the DARPA DC program.

Silverberg thanks PARC, the Stanford University Mathematics Department, and NSF (grant DMS-9988869).

abelian varieties and "trivial" cases. This suggests that genuinely new techniques might be necessary to construct multilinear maps with the desired properties.

2. Notation and Definitions

We first recall some standard notation and definitions that will be used throughout the paper.

(1) The set of all finite length binary strings is denoted $\{0,1\}^*$, and the set of all binary strings of length m is denoted $\{0,1\}^m$.
(2) We view a *randomized algorithm* $\mathcal{A}$ as a function on two inputs, $\mathcal{A}(x,r)$, where $x \in \{0,1\}^*$ is the input given to the algorithm and r is in $\{0,1\}^m$ for some m. Here r represents the sequence of random bits used by the algorithm. For an input x we let $\mathcal{A}(x)$ denote the random variable $\mathcal{A}(x,R)$, where R is uniformly distributed in $\{0,1\}^m$.
(3) The probability of an event $\mathcal{D}$ is denoted $\Pr[\mathcal{D}]$. For a finite set S we use $x \leftarrow S$ to define a random variable x that picks an element of S uniformly at random (that is, for all $c \in S$ we have $\Pr[x = c] = 1/|S|$). For a randomized algorithm $\mathcal{A}$ we use $x \leftarrow \mathcal{A}(y)$ to define a random variable x that is the output of algorithm $\mathcal{A}$ on input y. In other words, for all $c \in \{0,1\}^*$ we have $\Pr[x = c] = \Pr[\mathcal{A}(y,r) = c]$. We let
$$\Pr[b(x) :\ x \leftarrow \mathcal{A}(y)]$$
denote the probability that $b(x)$ is true, where x is the random variable defined by $x \leftarrow \mathcal{A}(y)$ and b is a predicate.
(4) We say that a function $\nu : \mathbb{Z}^+ \to \mathbb{R}^+$ is *negligible* if for all $d > 0$ and sufficiently large n we have $0 < \nu(n) < 1/n^d$. For example, $\nu(n) = 1/2^n$ is a negligible function.
(5) A function $f(n) : \mathbb{Z}^+ \to \mathbb{R}^+$ is *super-polynomial* if for all $c > 0$ and all sufficiently large n we have $f(n) \geq n^c$. A function $f(n) : \mathbb{Z}^+ \to \mathbb{R}^+$ is *super-linear* if for all $c > 0$ and all sufficiently large n we have $f(n) \geq cn$.

Next, we give a definition of an n-multilinear map. We view the groups G_1 and G_2 as multiplicative groups.

Definition 2.1. We say that a map $e : G_1^n \to G_2$ is an *n-multilinear map* if it satisfies the following properties:

(1) G_1 and G_2 are groups of the same prime order;
(2) if $a_1, \dots, a_n \in \mathbb{Z}$ and $x_1, \dots, x_n \in G_1$ then
$$e(x_1^{a_1}, \dots, x_n^{a_n}) = e(x_1, \dots, x_n)^{a_1 \cdots a_n};$$
(3) The map e is non-degenerate in the following sense: if $g \in G_1$ is a generator of G_1 then $e(g, ..., g)$ is a generator of G_2.

Let G_1, G_2 be finite cyclic groups of order ℓ and let g be a generator of G_1. Recall that the *discrete log* function in G_1 is defined as $\mathrm{Dlog}_g(g^\alpha) = \alpha$, where $\alpha \in \mathbb{Z}$ and $1 \leq \alpha \leq \ell$. The discrete log problem in G_1 is to compute the discrete log function in G_1. We are mostly interested in groups where this problem is intractable. It is well known [**26**] that computing discrete log in G_1 is reducible to computing discrete log in all prime order subgroups of G_1. Therefore, we can and will restrict our attention to groups G_1, G_2 of *prime* order ℓ.

We note that an efficiently computable n-multilinear map $e : G_1^n \to G_2$ can be used to reduce the discrete log problem in G_1 to the discrete log problem in G_2 (see [**19**] and [**10**]). Let $g, h \in G_1$ such that $h = g^\alpha$. Computing α given g and h is a discrete log problem in G_1. To reduce this to a discrete log problem in G_2 compute the following two values:

$$x = e(g, g, \ldots, g) \qquad \text{and} \qquad y = e(h, g, g, \ldots, g).$$

Then by n-multilinearity we have that $y = x^\alpha$ as elements in G_2. This simple argument shows that if the discrete log problem in G_1 is hard then discrete log in G_2 must also be hard. The converse is not known to be true.

Next, our goal is to define a *cryptographic* n-multilinear map generator. Roughly speaking, a cryptographic n-multilinear map $e : G_1^n \to G_2$ is an n-multilinear map such that (1) the group action in G_1 and G_2 is efficiently computable, (2) the map e is efficiently computable, and (3) there is no efficient algorithm to compute discrete log in G_1.

We will first define a multilinear map generator. Since we are studying computational problems on the groups G_1 and G_2 we cannot treat these groups as abstract algebraic objects. Instead, we have to fix an explicit representation of group elements and have to ensure that all group operations and n-multilinear maps are computable by polynomial time algorithms. Throughout the paper we represent group elements as binary strings, namely $G_1, G_2 \subset \{0,1\}^*$.

Definition 2.2. An *n-multilinear map description* $\Gamma \in \{0,1\}^*$ is a description of two groups G_1 and G_2 of the same prime order, an n-multilinear map $e : G_1^n \to G_2$, and functions prod_b, $\mathsf{inverse}_b$, map, and test_b, for $b = 1, 2$, satisfying:

- If $b = 1, 2$ and $x, y \in G_b$, then $\mathsf{prod}_b(\Gamma, x, y) = xy$ and $\mathsf{inverse}_b(\Gamma, x) = x^{-1}$.
- If $x_1, \ldots, x_n \in G_1$, then $\mathsf{map}(\Gamma, x_1, \ldots, x_n) = e(x_1, \ldots, x_n)$.
- If $b = 1, 2$ and $x \in \{0,1\}^*$, then $\mathsf{test}_b(\Gamma, x) = \mathsf{yes}$ if and only if $x \in G_b$.

For example, a 2-multilinear map description Γ might include a prime power q, coefficients for equations that define an abelian variety (or elliptic curve) A defined over $\mathbb{F}_q$, and the coordinates of a point $P \in A(\mathbb{F}_q)$ of prime order ℓ. The group G_1 would be the group generated by P, and G_2 would be the group of ℓ-th roots of unity in $\mathbb{F}_{q^d}^*$, where d is the order of $q \pmod{\ell}$. The map $e : G_1^2 \to G_2$ could be a modified Weil pairing (as in §7 below).

Definition 2.3. A *multilinear map generator* $\mathcal{G} = \mathcal{G}(t, n)$ is a randomized algorithm that runs in polynomial time in (positive integer) inputs t and n, and outputs a tuple (Γ, g, ℓ). Here Γ is an n-multilinear map description where (1) the functions prod_b, $\mathsf{inverse}_b$, map, and test_b run in polynomial time in t and n, (2) ℓ is the order of the groups G_1 and G_2 defined by Γ, and (3) g is some generator of G_1.

The point of the *security parameter* t in Definition 2.3 will become apparent when we define cryptographic multilinear map generators below. This parameter will determine the size of the groups G_1 and G_2. The size of G_1 as a function of t must be large enough so that no polynomial time algorithm in t can compute discrete log in G_1.

Let $\mathcal{G}$ be a multilinear map generator. Define a randomized algorithm $\mathcal{A}$'s *advantage* in computing discrete log to be the probability that $\mathcal{A}$ is able to compute discrete log in the group $G_1 = \langle g \rangle$ defined by $\mathcal{G}(t, n)$. In other words,

$$\mathsf{AdvDlog}_{\mathcal{G},\mathcal{A},n}(t) = \Pr\left[\mathcal{A}\left(\Gamma, g, g^r\right) = r \;:\; (\Gamma, g, \ell) \leftarrow \mathcal{G}(t, n),\; r \leftarrow \mathbb{Z}/\ell\mathbb{Z}\right].$$

Definition 2.4. A multilinear map generator $\mathcal{G}$ is a *cryptographic multilinear map generator* if for all polynomial time algorithms $\mathcal{A}$ (polynomial in t) and all $n > 1$, the function $\mathsf{AdvDlog}_{\mathcal{G},\mathcal{A},n}(t)$ is negligible.

Open problem. The central open problem posed in this paper is the construction of cryptographic multilinear map generators when $n > 2$.

For $n = 2$, (modified) Weil and Tate pairings on elliptic curves are believed to give cryptographic bilinear map generators. The constructions in this paper typically need n on the order of 160. Recall that asymptotically we only require that the n-multilinear map be computable in polynomial time in n (and t).

3. Complexity assumptions

For some of the applications we present, the intractability of discrete log is not sufficient to prove security. We will need to make slightly stronger assumptions. We list these assumptions here. The reader may wish to skip this section for now and refer back to it as needed in the later sections. For the remainder of this section, fix a multilinear map generator $\mathcal{G}$.

The multilinear Diffie-Hellman assumption. This assumption says that given $g, g^{a_1}, \ldots, g^{a_{n+1}}$ in G_1, it is hard to compute $e(g, \ldots, g)^{a_1 \cdots a_{n+1}}$ in G_2. More precisely, define a randomized algorithm $\mathcal{A}$'s advantage in solving the multilinear Diffie-Hellman problem to be the probability that $\mathcal{A}$ is able to compute

$$e(g, \ldots, g)^{a_1 \cdots a_{n+1}}$$

from $g, g^{a_1}, \ldots, g^{a_{n+1}}$, i.e.,

$$\begin{aligned}\mathsf{AdvDHm}_{\mathcal{G},\mathcal{A},n}(t) = \Pr[\mathcal{A}(\Gamma, g, g^{a_1}, \ldots, g^{a_{n+1}}) = e(g, \ldots, g)^{a_1 \cdots a_{n+1}} :\\ (\Gamma, g, \ell) \leftarrow \mathcal{G}(t, n),\ (a_1, \ldots, a_{n+1}) \leftarrow (\mathbb{Z}/\ell\mathbb{Z})^{n+1}].\end{aligned}$$

Definition 3.1. We say the multilinear map generator $\mathcal{G}$ satisfies the *multilinear Diffie-Hellman assumption* if for all polynomial time algorithms $\mathcal{A}$ (polynomial in t) and all $n > 1$, the function $\mathsf{AdvDHm}_{\mathcal{G},\mathcal{A},n}(t)$ is negligible.

The Diffie-Hellman inversion assumption. The assumption says that given $g, g^b \in G_1$ it is hard to compute $e(g, \ldots, g)^{1/b} \in G_2$. Define a randomized algorithm $\mathcal{A}$'s advantage in solving the Diffie-Hellman inversion problem to be the probability that $\mathcal{A}$ is able to compute $e(g, \ldots, g)^{1/b}$ from g, g^b, i.e.,

$$\begin{aligned}&\mathsf{AdvDHinv}_{\mathcal{G},\mathcal{A},n}(t) =\\ &\qquad \Pr\left[\mathcal{A}(\Gamma, g, g^b) = e(g, g, \ldots, g)^{1/b}\ :\ (\Gamma, g, \ell) \leftarrow \mathcal{G}(t, n),\ b \leftarrow (\mathbb{Z}/\ell\mathbb{Z})^{n+1}\right].\end{aligned}$$

Definition 3.2. The multilinear map generator $\mathcal{G}$ satisfies the *Diffie-Hellman inversion assumption* if for all polynomial time algorithms $\mathcal{A}$ (polynomial in t) and all $n > 1$, the function $\mathsf{AdvDHinv}_{\mathcal{G},\mathcal{A},n}(t)$ is negligible.

The generalized Diffie-Hellman assumption. The assumption says that given $g^{a_1}, \ldots, g^{a_n}$ in G_1 and given all the subset products $g^{\prod_{i \in S} a_i} \in G_1$ for any strict subset $S \subset \{1, \ldots, n\}$, it is hard to compute $g^{a_1 \cdots a_n} \in G_1$. Since the number of subset products is exponential in n we provide access to all these subset products through an oracle (an oracle is a function that can be evaluated in unit time). Let

(Γ, g, ℓ) be an output of $\mathcal{G}(t,n)$. For a vector $\vec{a} = (a_1, \ldots, a_n) \in (\mathbb{Z}/\ell\mathbb{Z})^n$, define $\mathcal{O}_{\Gamma,g,\vec{a}}$ to be an oracle that for any strict subset $S \subset \{1, \ldots, n\}$ responds with:

$$\mathcal{O}_{\Gamma,g,\vec{a}}(S) = g^{\prod_{i \in S} a_i} \in G_1.$$

Define a randomized algorithm $\mathcal{A}$'s advantage in solving the generalized Diffie-Hellman problem to be the probability that $\mathcal{A}$ is able to compute $g^{a_1 \cdots a_n}$ given access to the oracle $\mathcal{O}_{\Gamma,g,\vec{a}}(S)$. In other words,

$$\begin{aligned}\mathsf{AdvDHgen}_{\mathcal{G},\mathcal{A},n}(t) = \Pr[\mathcal{A}^{\mathcal{O}_{\Gamma,g,\vec{a}}}(\Gamma, g) = g^{a_1 \cdots a_n} :& \\ (\Gamma, g, \ell) \leftarrow \mathcal{G}(t,n),\ \vec{a} = (a_1, \ldots, a_n) \leftarrow (\mathbb{Z}/\ell\mathbb{Z})^n&].\end{aligned}$$

Note that the oracle only answers queries for strict subsets of $\{1, \ldots, n\}$.

Definition 3.3. We say $\mathcal{G}$ satisfies the *generalized Diffie-Hellman assumption* if for all polynomial time algorithms $\mathcal{A}$ (polynomial in t) and all $n > 1$, the function $\mathsf{AdvDHgen}_{\mathcal{G},\mathcal{A},n}(t)$ is negligible.

4. One-Round n-way Diffie-Hellman Key Exchange

We give several applications of n-multilinear maps to cryptography. We start with a simple application: constructing a one-round n-way Diffie-Hellman key exchange protocol. Joux [**15**] showed how Weil and Tate pairings can be used for a one-round 3-way secret key exchange. Using an n-multilinear map, Joux's protocol generalizes naturally to a one-round $(n+1)$-way secret key exchange.

Consider $n+1$ parties who wish to set up a conference key using a one-round protocol. The "one-round" refers to the fact that each party is only allowed to broadcast one value to all other parties. All $n+1$ broadcasts occur simultaneously. Once all $n+1$ parties broadcast their values, each party should be able to locally compute a global shared secret S. The secret S will then be used to derive a conference key. An eavesdropper, seeing only the public broadcast values, should not be able to compute the global secret S. This is a direct generalization of the Diffie-Hellman protocol to $n+1$ parties (Diffie-Hellman is designed for two parties). Solutions to this problem are useful in reducing the number of round trips in group key management protocols [**30**]. This is a long-standing open problem.

More precisely, a one-round n-way conference key exchange scheme consists of the following three randomized polynomial time algorithms:

$\mathsf{Setup}(t, n)$**:** Takes a security parameter $t \in \mathbb{Z}^+$ and the number of participants n. It runs in polynomial time in t, n and outputs public parameters $\Gamma_{dh} \in \{0,1\}^*$.

$\mathsf{Publish}(\Gamma_{dh}, i)$**:** Given an input $i \in \{1, \ldots, n\}$, the algorithm outputs a pair $(\mathrm{pub}_i, \mathrm{priv}_i)$, with both in $\{0,1\}^*$. Party i broadcasts pub_i to all other parties, and keeps priv_i secret.

$\mathsf{KeyGen}(\Gamma_{dh}, j, \mathbf{priv}_j,\ \{\mathbf{pub}_i\}_{i \neq j})$**:** Party $j \in \{1, \ldots, n\}$ collects the public broadcasts sent by all other parties. It then runs algorithm KeyGen giving it all these public values and its secret value priv_j. Algorithm KeyGen outputs a conference key S.

The consistency requirement is that for all $j = 1, \ldots, n$, algorithm KeyGen produces the same conference key S. In other words, all n parties generate the same secret conference key. The scheme is secure if no polynomial time algorithm, given all n public values $(\mathrm{pub}_1, \ldots, \mathrm{pub}_n)$, will produce the secret conference key S with non-negligible probability.

Definition 4.1. A one-round n-way conference key exchange scheme $\{\mathsf{Setup}$, $\mathsf{Publish}$, $\mathsf{KeyGen}\}$ is secure if for all polynomial time randomized algorithms $\mathcal{A}$ the following probability:

$$\begin{aligned}\mathsf{AdvDH}_{\mathcal{A},n}(t) = \Pr\big[&\mathcal{A}(\Gamma_{dh}, \mathrm{pub}_1, \ldots, \mathrm{pub}_n) = S \;:\; \Gamma_{dh} \leftarrow \mathsf{Setup}(t, n),\\ &(\mathrm{pub}_i, \mathrm{priv}_i) \leftarrow \mathsf{Publish}(\Gamma, i),\; S \leftarrow \mathsf{KeyGen}(\Gamma, 1, \mathrm{priv}_1, \{\mathrm{pub}_i\}_{i \neq 1})\big]\end{aligned}$$

is a negligible function in t.

We present a one-round $(n+1)$-way key exchange protocol from an n-multilinear map generator $\mathcal{G}$.

$\mathsf{Setup}(t, n+1)$**:** Run algorithm $\mathcal{G}(t, n)$ to get (Γ, g, ℓ). Let $e : G_1^n \rightarrow G_2$ be the n-multilinear map defined by Γ. Then g is a generator of G_1 and ℓ is the order of G_1. Output $\Gamma_{dh} = (\Gamma, g, \ell)$ as the public parameters.

$\mathsf{Publish}(\Gamma_{dh}, i)$**:** Pick a random integer $a_i \in [1, \ell-1]$. Compute $h_i = g^{a_i} \in G_1$. Output $(\mathrm{pub}_i, \mathrm{priv}_i)$ where $\mathrm{pub}_i = h_i$ and $\mathrm{priv}_i = a_i$.
Party i broadcasts h_i to all other participants and keeps a_i secret.

$\mathsf{KeyGen}(\Gamma_{dh}, j, \mathrm{priv}_j, \{\mathrm{pub}_i\}_{i \neq j})$: Let $\mathrm{priv}_j = a_j$ and $\mathrm{pub}_i = h_i$.
Party j computes the conference key S as follows:

$$S = e(h_1, \ldots, h_{j-1}, h_{j+1}, \ldots, h_{n+1})^{a_j} \in G_2.$$

This S is the output of algorithm KeyGen given $(\Gamma_{dh},\, j,\, \mathrm{priv}_j,\, \{\mathrm{pub}_i\}_{i \neq j})$ as input.

Note that $S = e(g, g, \ldots, g)^{a_1 a_2 \cdots a_{n+1}}$. Hence, all $n+1$ parties will obtain the same conference key S. The following result is immediate from Definition 3.1.

Proposition 4.2. *Let $\mathcal{G}$ be a multilinear map generator. If $\mathcal{G}$ satisfies the multilinear Diffie-Hellman assumption then the protocol above is a secure one-round $(n+1)$-way conference key exchange scheme for every $n > 1$.*

We note that to use the global secret S as a key for a symmetric cipher one would have to prove that S can be converted into a binary string of a certain length that is indistinguishable from a random string of the same length. This would require a stronger complexity assumption than the multilinear Diffie-Hellman assumption. Alternatively, one could use hard-core bits of e to generate the global secret one bit at a time, but this would require many invocations of the key exchange protocol above. This issue is analogous to the issue that comes up when using the standard Diffie-Hellman secret as a secret encryption key [**2**].

5. Unique Signatures and Proofs for the n-way Diffie-Hellman Relation

Our next application is useful for building unique signatures and verifiable pseudo random functions (VRF's) [**20**]. Let G_1 be a group of prime order ℓ with a generator g.

Definition 5.1. We say that $(g, g_1, \ldots, g_n, h) \in G_1^{n+2}$ is an *n-way Diffie-Hellman tuple* if g generates G_1 and there exist integers $a_1, \ldots, a_n \in [0, \ell-1]$ such that $g_i = g^{a_i}$ and $h = g^{a_1 \cdots a_n}$.

Suppose there is no efficient algorithm for the discrete log problem in G_1. We study the following problem: is there an efficient algorithm $\mathcal{A}$ that takes an arbitrary tuple $I = (g, g_1, \ldots, g_n, h) \in G_1^{n+2}$ as input and returns `yes` if and only if I is an n-way Diffie-Hellman tuple? We call this the n-way decision Diffie-Hellman problem. For $n = 2$ one obtains the standard Decision Diffie-Hellman problem [**2**].

Recently Joux and Nguyen [**16**] showed that the group of points on a supersingular elliptic curve over a finite field is an example of a group where discrete log is (presumably) hard, but the standard (2-way) Decision Diffie-Hellman problem is easy. A generalization of their idea using an n-multilinear map solves the n-way decision Diffie-Hellman problem.

ALGORITHM 5.2. Suppose $e : G_1^n \to G_2$ is an n-multilinear map. Let g be a generator of G_1 and let $I = (g, g_1, \ldots, g_n, h) \in G_1^{n+2}$. We test if I is an n-way Diffie-Hellman tuple as follows:

(1) Compute $A = e(g_1, \ldots, g_n) \in G_2$.
(2) Compute $B = e(h, g, g, \ldots, g) \in G_2$.
(3) Test if $A = B$. If so, output `yes`. If not, output `no`.

The following simple result shows that the algorithm's output is always correct.

PROPOSITION 5.3. *Suppose that $e : G_1^n \to G_2$ is an n-multilinear map, and $I = (g, g_1, \ldots, g_n, h) \in G_1^{n+2}$, where g is a generator of G_1. Algorithm 5.2 outputs* `yes` *given I as input if and only if I is an n-way Diffie-Hellman tuple.*

PROOF. Write $g_i = g^{a_i}$ and $h = g^b$. Then

$$\begin{aligned} e(g_1, \ldots, g_n) &= e(g^{a_1}, \ldots, g^{a_n}) = e(g, \ldots, g)^{a_1 \cdots a_n}, \\ e(h, g, g, \ldots, g) &= e(g^b, g, g, \ldots, g) = e(g, g, \ldots, g)^b. \end{aligned}$$

The non-degeneracy of e implies that $e(g, g, \ldots, g)$ is a generator of G_2. It now follows that $e(g_1, \ldots, g_n) = e(h, g, g, \ldots, g)$ if and only if $b \equiv a_1 \cdots a_n \pmod{\ell}$. $\square$

We have just shown that a cryptographic n-multilinear map generator would give rise to groups where discrete log is hard, but the n-way decision Diffie-Hellman problem is easy.

5.1. Unique Signatures and Verifiable Pseudo Random Functions. Using Algorithm 5.2 we give a simple construction for a unique signature scheme and Verifiable Pseudo Random Functions. We first recall the definition of unique signatures [**12**]. Intuitively, a unique signature scheme is a digital signature scheme where every message has a unique digital signature (in most secure signature schemes there are many valid signatures for a given message). Unique signature schemes were known to exist in the common random string model [**12**] and in the random oracle model [**1**], but until the results of Micali et al. [**20**] there were no constructions for such schemes in the standard model defined below. Unique signatures are used to construct Verifiable Pseudo Random Functions, which are a useful tool in cryptographic protocol design [**20**].

DEFINITION 5.4. An *n-bit unique signature scheme* (which is used to sign n-bit messages) consists of three algorithms $\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify}$ defined as follows:

$\mathsf{KeyGen}(t)$**:** A randomized algorithm that outputs a signing key $\mathsf{S}K$ and a verification key $\mathsf{V}K$.

$\mathsf{Sign}(M, \mathsf{S}K)$**:** A deterministic algorithm that takes as input a message $M \in \{0,1\}^n$ and a signing key $\mathsf{S}K$ and outputs a signature S.

$\mathsf{Verify}(M, S, \mathsf{V}K)$**:** A deterministic algorithm that takes as input a message $M \in \{0,1\}^n$, a signature S, and a verification key $\mathsf{V}K$ and outputs yes or no.

These algorithms must satisfy the following requirements:

Consistency: For every key pair $(\mathsf{V}K, \mathsf{S}K)$ produced by the KeyGen algorithm and every message $M \in \{0,1\}^n$ we have that

$$\mathsf{Verify}(M, \mathsf{Sign}(M, \mathsf{S}K), \mathsf{V}K) = \mathsf{yes}.$$

Uniqueness: For every key pair $(\mathsf{V}K, \mathsf{S}K)$ produced by the KeyGen algorithm, every message $M \in \{0,1\}^n$, and every S_1 and S_2, we have that

$$\mathsf{Verify}(M, S_1, \mathsf{V}K) = \mathsf{Verify}(M, S_2, \mathsf{V}K) = \mathsf{yes} \quad \Rightarrow \quad \mathsf{S}_1 = \mathsf{S}_2.$$

Security for a unique signature scheme is defined as for standard signatures and is called *security against existential forgery under an adaptive chosen message attack* [**11**]. This notion is defined by the following game between a challenger and an attacker $\mathcal{A}$:

Step 1: The challenger runs algorithm $\mathsf{KeyGen}(t)$ to generate a key pair $(\mathsf{V}K, \mathsf{S}K)$. It gives $\mathsf{V}K$ to the attacker and keeps $\mathsf{S}K$ to itself.

Step 2: The attacker $\mathcal{A}$ issues finitely many queries $M_1, M_2, \ldots$ in $\{0,1\}^n$ and receives the signatures $S_1, S_2, \ldots$ on these queries. These queries can be issued adaptively, namely, the attacker can choose query M_i after seeing the signatures $S_1, \ldots, S_{i-1}$.

Step 3: Finally, the attacker $\mathcal{A}$ outputs a message signature pair (M, S) where $M \notin \{M_1, M_2, \ldots\}$.

The attacker $\mathcal{A}$ wins the game if $\mathsf{Verify}(M, S, \mathsf{V}K) = \mathsf{yes}$. Let $\mathsf{AdvSig}_{Sig,\mathcal{A}}(t)$ denote the probability that $\mathcal{A}$ wins the game.

DEFINITION 5.5. We say that an n-bit unique signature scheme Sig is secure against existential forgery under an adaptive chosen message attack if for all polynomial time attack algorithms $\mathcal{A}$ (polynomial in t) the function $\mathsf{AdvSig}_{Sig,\mathcal{A}}(t)$ is negligible.

We give a simple construction for unique signatures. The construction is similar to a Pseudo Random Function (PRF) based on the Decision Diffie-Hellman problem (DDH) due to Naor and Reingold [**25**]. Our construction is based on a recent result due to Lysyanskaya [**17**] who proposed a unique signature scheme where the signature on an n-bit message consists of n group elements. We show that multilinear maps give rise to a signature scheme where a signature on an n-bit message is a single group element.

Let $\mathcal{G}$ be a multilinear map generator. The following unique signature scheme is used to sign n-bit messages:

$\mathsf{KeyGen}(t)$**:** (1) Run algorithm $\mathcal{G}(t, n)$ to generate (Γ, g, ℓ).

(2) Pick random $a_{1,0}, a_{1,1}, \ldots, a_{n,0}, a_{n,1} \in \{1, \ldots, \ell - 1\}$.

(3) Set the signing key $\mathsf{S}K = (\Gamma, a_{1,0}, a_{1,1}, \ldots, a_{n,0}, a_{n,1})$, and the verification key $\mathsf{V}K = (\Gamma, g, g^{a_{1,0}}, \ldots, g^{a_{n,1}})$.

$\mathsf{Sign}(M, \mathsf{S}K)$**:** Let $M = m_1 \ldots m_n \in \{0,1\}^n$. Output:

$$S = g^{a_{1,m_1} \cdot a_{2,m_2} \cdots a_{n,m_n}} \quad \in G_1.$$

$\mathsf{Verify}(M, S, \mathsf{V}K)$**:** Write $\mathsf{V}K = (\Gamma, g, g_{1,0}, \ldots, g_{n,1})$.
Test if $I = (g, g_{1,m_1}, \ldots, g_{n,m_n}, S)$ is an n-way Diffie-Hellman tuple using Algorithm 5.2. Output yes if I is an n-way Diffie-Hellman tuple and output no otherwise.

For a given M and $\mathsf{V}K$ there is only one $S \in G_1$ for which

$$(g, g_{1,m_1}, \ldots, g_{n,m_n}, S)$$

is an n-way Diffie-Hellman tuple. Hence, the scheme is a unique signature scheme. Note that there is some negligible probability that two different messages have the same signature.

Next, we argue that the scheme is a secure unique signature scheme. Security is based on the generalized Diffie-Hellman assumption (Definition 3.3).

THEOREM 5.6. *Suppose the multilinear map generator $\mathcal{G}$ satisfies the generalized Diffie-Hellman assumption. Then for all fixed $n \in \mathbb{Z}^+$, the n-bit unique signature scheme above is secure against existential forgery under an adaptive chosen ciphertext attack. Concretely, an attack algorithm $\mathcal{A}$ with advantage $\mathsf{AdvSig}_{Sig,\mathcal{A}}(t)$ in forging signatures gives rise to an algorithm $\mathcal{B}$ for the generalized Diffie-Hellman problem in $\mathcal{G}$ with advantage*

$$\mathsf{AdvDHgen}_{\mathcal{G},\mathcal{B},n}(t) \geq \mathsf{AdvSig}_{Sig,\mathcal{A}}(t)/2^n.$$

PROOF. The proof is essentially identical to the proof of security given by Lysyanskaya [**17**]. □

Concrete parameters. Signature schemes in practice are mostly used to sign short messages that are the output of a collision resistant hash function such as SHA-1. Using the terminology above, to sign a message M of arbitrary length we compute $S = \mathsf{Sign}(H(M), \mathsf{S}K)$ where H is some collision resistant hash. Therefore, by Theorem 5.6, if H outputs n-bit strings then we need an n-multilinear map generator $\mathcal{G}$ for which $2^n \mathsf{AdvDHgen}_{\mathcal{G},\mathcal{B},n}(t)$ is negligible. In practice we often use $n = 160$ since the output of SHA-1 is 160-bit strings. Thus to give concrete parameters, we need a group $G_1 = \langle g \rangle$ where the generalized Diffie-Hellman problem cannot be solved in time 2^{80} with advantage greater than $1/2^{240}$ (this will ensure that no 2^{80}-time algorithm can existentially forge signatures with probability greater than $1/2^{80}$). It is currently (believed to be) possible to build groups where the Diffie-Hellman problem cannot be solved in time 2^{80} with advantage greater than $1/2^{240}$ (using groups of points on elliptic curves over sufficiently large finite fields). We hope that a 160-multilinear map $e : G_1^{160} \to G_2$ can be built for which G_1 has the same security parameters for the generalized Diffie-Hellman problem. We note that, as in [**17**], the reduction in Theorem 5.6 can be made more efficient by restricting the message space to codewords in a certain error correcting code.

Signature length. Note that a signature in the scheme above consists of a single group element in G_1. This means that this signature scheme can potentially produce signatures that are as short as BLS signatures [**4**]. BLS signatures are existentially unforgeable in the random oracle model, whereas the advantage of the signature scheme above is that it is existentially unforgeable in the standard security model (no random oracles are needed).

To conclude the section we note that Micali et al. [**20**] show that unique signatures give rise to Verifiable Pseudo Random Functions (VRF). Hence, the construction using n-multilinear maps also gives a simple construction for VRF's.

6. Broadcast Encryption with Short Keys and Transmissions

Broadcast encryption [**8**] appears to be the most interesting application to date for n-multilinear maps. We begin by describing the broadcast encryption problem, survey some of the existing work, and then describe a solution using n-multilinear maps.

6.1. The broadcast encryption problem. Broadcast encryption involves one broadcaster and n receivers. Each receiver is given a unique private key. The broadcaster is given a broadcaster key. The broadcaster wishes to broadcast messages to a specific subset $S \subseteq \{1, \dots, n\}$ of receivers (say, those receivers that previously paid to receive the broadcast). Any receiver in S should be able to use its private key to decrypt the broadcast. However, even if all receivers outside of S collude they should not be able to decrypt the broadcast. More precisely, a broadcast encryption scheme is made up of three randomized polynomial time (in t and n) algorithms:

$\mathsf{Setup}(t, n)$**:** Takes as input a security parameter $t \in \mathbb{Z}^+$ and the number of receivers n. It outputs n private keys $d_1, \dots, d_n$ and a sender key T.

$\mathsf{Encrypt}(S, T)$**:** Takes as input a subset $S \subseteq \{1, \dots, n\}$, and sender key T. It outputs a pair (Hdr, K) where Hdr is called the header and K is a message encryption key. Let C_M be the encryption of the message body M under the symmetric key K. The broadcast to users consists of (S, Hdr, C_M). The pair (S, Hdr) is often called the full header and C_M is often called the broadcast body.

$\mathsf{Decrypt}(S, d_i, \mathrm{Hdr})$**:** Takes as input a subset $S \subseteq \{1, \dots, n\}$, a receiver key d_i, and a header Hdr. If $i \in S$, then the algorithm outputs the message encryption key K. The key K can then be used to decrypt C_M and obtain the message body M.

To state a (simple) security requirement we define the following game between an attack algorithm $\mathcal{A}$ and a challenger.

Step 1: The challenger takes (t, n) as input, and runs $\mathsf{Setup}(t, n)$ to generate a sender key T and n private keys $d_1, \dots, d_n$.

Step 2: Algorithm $\mathcal{A}$ outputs a set $S \subseteq \{1, \dots, n\}$ of receivers where it wants to mount an attack. The challenger gives $\mathcal{A}$ all private keys d_j for which $j \notin S$.

Step 3: The challenger runs the Encrypt algorithm to obtain $(\mathrm{Hdr}, K) = \mathsf{Encrypt}(S, T)$. It gives Hdr to algorithm $\mathcal{A}$.

Step 4: Algorithm $\mathcal{A}$ outputs a key K' and wins the game if $K = K'$.

Let $\mathsf{AdvBr}_{\mathcal{A},n}(t)$ denote the probability that $\mathcal{A}$ wins the game when the challenger is given (t, n) as input.

Observe that this game models an attack where all users not in the set S collude to try and expose a broadcast intended for users in S only. The set S is chosen adversarially.

Definition 6.1. We say that the broadcast encryption scheme is secure if for all polynomial time attack algorithms $\mathcal{A}$ and for all $n > 1$ the function $\mathsf{AdvBr}_{\mathcal{A},n}(t)$ is negligible.

Note that in the attack game above the adversary is non-adaptive — it requests the entire set of keys S at once. An adaptive adversary could request user keys adaptively. That is, it would decide to request the private key for user i_r after seeing the private keys for users $i_1, i_2, \ldots, i_{r-1}$. Here we only consider non-adaptive adversaries.

The question is how to build broadcast encryption schemes where both the header size and private key size are small as a function of the number n of receivers. One trivial construction gives a secure broadcast encryption scheme where the size of the private keys d_i is independent of n, but unfortunately the header size is linear in n. Another trivial construction gives a broadcast encryption scheme where the size of the header Hdr is independent of n, but the size of each private key d_i is exponential in n. These are two extremes of the spectrum. Recently Naor-Naor-Lotspiech [**24**] gave an elegant construction where each private key consists of $O((\log n)^2)$ encryption keys for a symmetric encryption scheme. The header consists of $O(n-|S|)$ encryptions of a message key using the symmetric encryption scheme. When the size of the symmetric encryption key is k-bits the system has the following parameters:

$$\text{private-key-size} = O(k(\log n)^2) \quad ; \qquad \text{header-size} = O(k(n-|S|)).$$

Halevi and Shamir [**13**] showed that the private key size can be reduced to approximately $O(k \log n)$. This broadcast system is designed to broadcast to large sets S, i.e., when the size of S is close to n, so that $n-|S|$ is small. The value of k depends on the security parameter t. We must ensure that a polynomial time algorithm (in t) cannot scan through the entire set of symmetric keys, a set of size 2^k. Therefore, for simplicity we say that k must be at least $c(\log t)^2$ for some constant $c > 0$. In fact, any super-linear function in $\log t$ will do. For consistency with the notation in this section we say that the scheme, with the improvement of Halevi-Shamir, has the following parameters:

$$\text{private-key-size} = O\left((\log t)^2 \log n\right) \quad ; \qquad \text{header-size} = O\left((\log t)^2 (n-|S|)\right).$$

A central open problem in this area is whether one can build a secure broadcast encryption scheme where both the size of the header and the size of each private key d_i depend at most logarithmically on n. We note that Fiat-Naor [**8**] and Chick-Tavaras [**6**] gave constructions based on RSA that meet this requirements. However, these constructions either do not resist collusion of users [**8**] outside the set S, or the construction can only handle a small number [**6**] of receiver sets S.

6.2. An efficient solution using n-multilinear maps. Using n-multilinear maps it is possible to give an efficient solution to the broadcast encryption problem (efficient in terms of private key size and header size). We construct a secure broadcast scheme with the following parameters:

$$\text{private-key-size} = O((\log t)^2) \quad ; \qquad \text{header-size} = 0.$$

In fact, $(\log t)^2$ can be replaced by any super-linear function in $\log t$.

Let $\mathcal{G}$ be a multilinear map generator and let n be the intended number of receivers. Let (Γ, g, ℓ) be an output of $\mathcal{G}(t, n)$. The order of G_1 must be sufficiently large to make discrete log difficult. We assume elements in G_1 are represented as binary strings of length $O((\log t)^2)$. Since we always assume $t > n$, the important point here is that the length of elements in G_1 depends at most logarithmically on n.

We will also fix a function $F_{m,\Gamma} : \{0,1\}^m \to G_1^n$. We call $\{0,1\}^m$ the seed space. We will need $m = m(t)$ to be a function of the security parameter t. The function $m(t)$ will be determined later.

For a given seed $a \in \{0,1\}^m$, a given set $S \subseteq \{1,\ldots,n\}$, and a given $g \in G_1$ we define an auxiliary function $\Phi_{S,a,g} : \{1,\ldots,n\} \to G_1$ as follows:

$$\Phi_{S,a,g}(i) = \begin{cases} g_i & \text{if } i \in S \\ g & \text{otherwise} \end{cases}$$

where $F_{m,\Gamma}(a) = (g_1,\ldots,g_n)$. We describe the new broadcast encryption scheme by describing the three algorithms Setup, Encrypt, and Decrypt.

$\mathsf{Setup}(t,n)$**:** Run algorithm $\mathcal{G}(t,n)$ to generate (Γ, g, ℓ).
Pick a random $\alpha \in [1, \ell-1]$.
Pick a random $a \in \{0,1\}^m$ and write $F_{m,\Gamma}(a) = (g_1,\ldots,g_n) \in G_1^n$.
The sender key is $T = (\Gamma, g, a, \alpha)$.
The i-th receiver key is $d_i = (i, \Gamma, g, a, u_i)$ where $u_i = g_i^\alpha$.

$\mathsf{Encrypt}(S,T)$**:** To transmit to a set S do:
Step 1: Compute $K_S = e\big(\Phi_{S,a,g}(1),\ldots,\Phi_{S,a,g}(n)\big)^\alpha \in G_2$.
Step 2: Output K_S as the message encryption key. The header Hdr is the empty string ε (in other words, the size of the header is zero).

$\mathsf{Decrypt}(S, d_i, \varepsilon)$**:** To obtain the message encryption key K_S using d_i, compute:

$$K_S = e\Bigg(\Phi_{S,a,g}(1),\ldots,\Phi_{S,a,g}(i-1),\ u_i,\ \Phi_{S,a,g}(i+1),\ldots,\Phi_{S,a,g}(n)\Bigg).$$

The security of the system relies on the Diffie-Hellman inversion assumption (Definition 3.2). We show that an attack on the broadcast encryption scheme leads to an algorithm that can solve the Diffie-Hellman inversion problem for $\mathcal{G}$. Unfortunately, the proof requires that the function $F_{m,\Gamma} : \{0,1\}^m \to G_1^n$ be modeled as a random oracle (see [**1**] for the definition; essentially, a random oracle implements a function chosen uniformly at random from the set of all functions from the domain to the range).

Theorem 6.2. *Suppose the multilinear map generator $\mathcal{G}$ satisfies the Diffie-Hellman inversion assumption, and suppose the function $F_{m,\Gamma} : \{0,1\}^m \to G_1^n$ is a random oracle. Then the broadcast encryption scheme above is secure as long as $m = m(t)$ is a super-linear function in $\log t$ (e.g., $m(t) = (\log t)^2$).*

Proof. Suppose there is a polynomial time attacker $\mathcal{A}$ that wins the broadcast encryption game with non-negligible probability, i.e., $\epsilon(t) = \mathsf{AdvBr}_{\mathcal{A},n}(t) > 1/t^c$ for some $c > 0$. Let $T(t)$ be the running time of algorithm $\mathcal{A}$. We know that $T(t) < t^d$ for some $d > 0$. We build an algorithm $\mathcal{B}$ for solving the Diffie-Hellman inversion problem where

$$\mathsf{AdvDHinv}_{\mathcal{G},\mathcal{B},n}(t) > \epsilon(t) - \frac{T(t)}{2^{m(t)} - T(t)}.$$

Since $m(t)$ is a super-linear function in $\log t$ we know that $2^{m(t)}$ is super-polynomial and therefore $\frac{T(t)}{2^{m(t)}-T(t)}$ is a negligible function. It follows that $\mathsf{AdvDHinv}_{\mathcal{G},\mathcal{B},n}(t)$ is non-negligible, and hence $\mathcal{B}$ will violate the Diffie-Hellman inversion assumption for $\mathcal{G}$.

We describe algorithm $\mathcal{B}$. Let $(\Gamma, g, \ell) \leftarrow \mathcal{G}(t,n)$. As usual, Γ defines an n-multilinear map $e : G_1^n \to G_2$. Algorithm $\mathcal{B}$ is given Γ and $g, h \in G_1$. Write $h = g^b$

where $b \in [1, \ell - 1]$. Algorithm $\mathcal{B}$'s goal is to construct $e(g, g, \ldots, g)^{1/b}$, the b-th root of $e(g, g, \ldots, g)$ in G_2.

Algorithm $\mathcal{B}(\Gamma, g, h)$ works by running $\mathcal{A}$ as follows:

$F_{m,\Gamma}$-queries: At any time algorithm $\mathcal{A}$ may query the oracle for the function $F_{m,\Gamma}$. To answer these queries $\mathcal{B}$ maintains an F-list consisting of tuples $(a, (g_1, \ldots, g_n))$. Initially the F-list is empty. When $\mathcal{A}$ issues a query for $F_{m,\Gamma}(a)$ with $a \in \{0,1\}^m$, algorithm $\mathcal{B}$ checks whether a appears as the first entry of some tuple $(a, (g_1, \ldots, g_n))$ on the F-list. If so, $\mathcal{B}$ replies with $F_{m,\Gamma}(a) = (g_1, \ldots, g_n)$. Otherwise, $\mathcal{B}$ picks a random tuple $(g_1, \ldots, g_n) \in G_1^n$, appends the tuple $(a, (g_1, \ldots, g_n))$ to the F-list, and responds to $\mathcal{A}$ with $F_{m,\Gamma}(a) = (g_1, \ldots, g_n)$.

Step 1: At the beginning of the attack game, algorithm $\mathcal{A}$ outputs a subset of users $S \subseteq \{1, \ldots, n\}$. Algorithm $\mathcal{B}$ needs to respond with all private keys for users $i \notin S$. It does so as follows:

(1) Pick a random $a \in \{0,1\}^m$. If a already appears as the first entry of some tuple on the F-list, algorithm $\mathcal{B}$ outputs `fail` and terminates the simulation. The algorithm failed.

(2) Otherwise, algorithm $\mathcal{B}$ picks random $r_1, \ldots, r_n \in \{1, \ldots, \ell\}$. For $i \in S$ set $g_i = g^{r_i}$. For $i \notin S$ set $g_i = h^{r_i}$. Let $(g_1, \ldots, g_n)$ be the resulting tuple. We define $F_{m,\Gamma}(a) = (g_1, \ldots, g_n)$ and append the tuple $(a, (g_1, \ldots, g_n))$ to the F-list.

(3) At this point we know that $F_{m,\Gamma}(a) = (g_1, \ldots, g_n)$. For $i \notin S$ define $d_i = (i, \Gamma, g, a, u_i)$ where $u_i = g^{r_i}$. Note that for all $i \notin S$ we have $u_i = g_i^{1/b}$. This means that the set of private keys $\{d_i\}_{i \notin S}$ is valid and consistent. The (unknown) secret α that would normally be used to generate these keys is defined to be $\alpha = b^{-1} \bmod \ell$.

(4) For all $i \notin S$ give d_i to algorithm $\mathcal{A}$.

Step 2: We know that algorithm $\mathcal{A}$ will respond with the key for the set S, namely:

$$K_S = e\Big(\Phi_{S,a,g}(1), \ldots, \Phi_{S,a,g}(n)\Big)^{\alpha} = e\Big(\Phi_{S,a,g}(1), \ldots, \Phi_{S,a,g}(n)\Big)^{\frac{1}{b}}$$

with probability at least $\epsilon(t)$. By definition we have

$$K_S = e(g, g, \ldots, g)^{b^{-1} \prod_{i \in S} r_i}.$$

Step 3: Set $c = \left(\prod_{i \in S} r_i\right)^{-1} \pmod{\ell}$. Then $(K_S)^c = e(g, g, \ldots, g)^{1/b}$. Hence, by computing K_S^c, algorithm $\mathcal{B}$ obtains the value it was asked to compute.

Algorithm $\mathcal{B}$ will produce the correct answer if (1) it does not abort in Step 1, and (2) it receives the correct answer from algorithm $\mathcal{A}$ in Step 2. By definition of algorithm $\mathcal{A}$ we know that event (2) happens with probability at least $\epsilon(t)$. To bound the probability for event (1) first observe that $\mathcal{A}$ makes at most $T(t)$ queries to the function $F_{m,\Gamma}$ prior to Step 1. Algorithm $\mathcal{B}$ will abort in Step 1 if it picks a random $a \in \{0,1\}^m$ that happens to equal one of $\mathcal{A}$'s queries. The probability that $\mathcal{A}$'s i-th query is equal to a given that the first $i-1$ queries are distinct and not equal to a is at most $\frac{1}{2^m - i}$. Hence, whenever $T(t) < 2^m$, the probability that

$\mathcal{B}$ aborts in Step 1 is at most

$$\frac{1}{2^m} + \frac{1}{2^m - 1} + \ldots + \frac{1}{2^m - T(t) + 1} < \frac{T(t)}{2^m - T(t)}.$$

Hence,

$$\mathsf{AdvDHinv}_{\mathcal{G},\mathcal{B},n}(t) = \Pr[(1) \text{ and } (2)] \geq \Pr[(2)] - \Pr[\neg(1)] \geq \epsilon(t) - \frac{T(t)}{2^m - T(t)}$$

as required. □

Summary of the parameters. Suppose $\mathcal{G}$ satisfies the Diffie-Hellman inversion assumption. Then by Theorem 6.2, to get a secure broadcast encryption scheme we can take $m = (\log t)^2$. The private key consists of an m-bit string and two group elements in G_1. By assumption, the two group elements are also of length $O(m)$. Hence, we get the following parameters for our scheme:

$$\text{private-key-size} = O((\log t)^2) \quad ; \quad \text{header-size} = 0.$$

The full header in the scheme contains only the description of the set S. Since we always assume that $t > n$ we get that the size of the private key depends logarithmically on the number of receivers n. Hence, multilinear maps give a broadcast encryption scheme with optimal size broadcast and very short private keys. On the down side, encryption and decryption take time proportional to n.

7. Constructions and Restrictions

Where does one look for n-multilinear forms with the desired properties?

For $n = 2$, the answer is Weil and Tate pairings associated to abelian varieties. If A is a principally polarized supersingular abelian variety over a finite field F, then the Weil pairing $\hat{e}_N$, for any positive integer N not divisible by the characteristic of F, is a Galois-equivariant non-degenerate bilinear map $\hat{e}_N : A[N] \times \hat{A}[N] \to \boldsymbol{\mu}_N$, where $A[N]$ is the N-torsion on A, $\hat{A}[N]$ is the N-torsion on the dual abelian variety, and $\boldsymbol{\mu}_N$ is the group of N-th roots of unity. A principal polarization then induces a map $e_N : A[N] \times A[N] \to \boldsymbol{\mu}_N$. When A is a Jacobian variety, [**21**] (see also Section 5.1 of [**18**] for the case of elliptic curves) gives an algorithm for computing the pairing. If $P \in A(F)$ is a point of prime order ℓ, and $\varphi \in \mathrm{End}(A)$ sends P to an independent point of order ℓ, then the modified Weil pairing $\hat{e} : G_1^2 \to G_2$ defined by $\hat{e}(P_1, P_2) = e_\ell(P_1, \varphi(P_2))$ is a 2-multilinear map in the sense of Section 3, where G_1 is the subgroup of $A(F)$ generated by P, and G_2 is the group of ℓ-th roots of unity. When A is a supersingular Jacobian variety, then the group G_2 lies in a relatively small extension of the ground field F, and therefore Miller's algorithm is efficient.

If K/F is a Galois extension, V and W are sets with a $\mathrm{Gal}(K/F)$-action, and $f : V \to W$ is a function, then f is called $\mathrm{Gal}(K/F)$-*equivariant* if for all $\sigma \in \mathrm{Gal}(K/F)$ and $x \in V$, we have $\sigma(f(x)) = f(\sigma(x))$. If A and B are algebraic varieties defined over a field F, and $f : A \to B$ is a morphism defined by polynomial equations with coefficients in F, then for every Galois extension K of F the induced map $f_K : A(K) \to B(K)$ is $\mathrm{Gal}(K/F)$-equivariant. If a map between algebraic varieties is computable, we would expect it to be defined by polynomial equations, i.e., to be algebraic, and therefore Galois-equivariant.

Very roughly speaking, a *motive* over a field is something whose "realizations" behave as if they were the cohomology groups associated to a variety. According to

3.1 of [**29**], "one reason for Grothendieck's introduction of motives was to serve as analogues of the Jacobian of a curve in higher dimensions." See [**14**] for a treatment of motives over finite fields. If varieties giving rise to n-multilinear maps cannot be found for $n > 2$, one could at least hope that such maps might arise from motives. The results below give evidence that even such a hope might be too optimistic. We believe that our paper is the first to give connections between the theory of motives and the field of cryptography. We expect that the motivic point of view will prove to be valuable in better understanding the mathematics that underlies public key cryptography.

If $e : G_1^n \to \boldsymbol{\mu}_\ell$ is an n-multilinear map where G_1 is a group of prime order ℓ that comes from geometry (or from a motive), then it might be reasonable to expect that the underlying geometric object or motive would in fact give rise to a compatible system of such maps that are Galois-equivariant, for all but finitely many primes ℓ. It is reasonable to expect such a map e to come about by restricting (to one-dimensional subspaces) a multilinear and Galois-equivariant map $V^n \to \boldsymbol{\mu}_\ell$, where V is a finite-dimensional $\mathbb{F}_\ell$-vector space with a Galois action, coming from the Galois action on the underlying geometric object (or the ℓ-adic realization of the motive). This is the case for Weil and Tate pairings on abelian varieties. (But note that, while the Weil pairings e_ℓ are all Galois-equivariant, the modified pairing $\hat{e}$ defined above only becomes Galois-equivariant after passing to a field where the endomorphism φ is defined.) We will give evidence that suggests that there is something special about pairings on abelian varieties that permits this to happen.

7.1. Preliminaries. We begin with some notation. If F is a field and F^s is a separable closure, let $G_F = \mathrm{Gal}(F^s/F)$. Suppose $N \in \mathbb{Z}^+$ and $\mathrm{char}(F) \nmid N$. Write $\boldsymbol{\mu}_N$ for the group of N-th roots of unity in F^s. The cyclotomic character

$$\chi_N : G_F \to (\mathbb{Z}/N\mathbb{Z})^\times$$

is defined by $\sigma(\zeta) = \zeta^{\chi_N(\sigma)}$ for every $\sigma \in G_F$ and $\zeta \in \boldsymbol{\mu}_N$.

REMARK 7.1. If $V_1, \ldots, V_n$ are finite-dimensional $\mathbb{F}_\ell$-vector spaces, then there is a natural one-to-correspondence between multilinear homomorphisms

$$h : V_1 \times \cdots \times V_n \to \boldsymbol{\mu}_\ell$$

and linear homomorphisms

$$\tilde{h} : V_1 \otimes \cdots \otimes V_n \to \boldsymbol{\mu}_\ell,$$

with $h(x_1, \ldots, x_n) = \tilde{h}(x_1 \otimes \ldots \otimes x_n)$.

LEMMA 7.2. *Suppose $F = \mathbb{F}_q$, N and d are positive integers,* $\mathrm{char}(F) \nmid N$, *and $\chi_N : G_F \to (\mathbb{Z}/N\mathbb{Z})^\times$ is the cyclotomic character. Then $\chi_N = \chi_N^d$ if and only if $q^{d-1} - 1$ is divisible by N.*

PROOF. We have $\chi_N = \chi_N^d \iff \chi_N^{d-1} = 1 \iff d-1$ is divisible by the order of χ_N, which is $[F(\zeta_N) : F]$ where ζ_N is a primitive N-th root of unity. Equivalently, $\zeta_N \in \mathbb{F}_{q^{d-1}}$, i.e., N divides $|\mathbb{F}_{q^{d-1}}| = q^{d-1} - 1$. $\square$

Note that when $d = 1$, the condition that N divide $q^{d-1} - 1$ is trivially true. As we will see below, it is this condition that makes n-multilinear forms special in the case where $n = 2$.

7.2. Tensor products of Weil pairings. We next discuss a straightforward generalization of the Weil pairing, namely a tensor product of Weil pairings. The linearity is now obvious, as is the Galois-equivariance as a map from $A[N]^{2r}$ to $\boldsymbol{\mu}_N^{\otimes r}$. However, after composing this map with an isomorphism $\boldsymbol{\mu}_N^{\otimes r} \to \boldsymbol{\mu}_N$, the resulting map from $A[N]^{2r}$ to $\boldsymbol{\mu}_N$ is Galois-equivariant if and only if $q^{r-1}-1$ is divisible by N, where q is the size of the finite field of definition.

More precisely, suppose that $F = \mathbb{F}_q$, that N and r are positive integers, that $\gcd(N,q) = 1$, that A is a principally polarized abelian variety over F, and that $e_N : A[N] \times A[N] \to \boldsymbol{\mu}_N$ is the Weil pairing induced by a principal polarization on A. Then the form $e_{r,N} : A[N]^{2r} \to \boldsymbol{\mu}_N^{\otimes r}$ defined by

$$e_{r,N}(P_1,\ldots,P_r,Q_1,\ldots,Q_r) = e_N(P_1,Q_1) \otimes \cdots \otimes e_N(P_r,Q_r)$$

is multilinear and G_F-equivariant.

If $r = 1$, then $e_{r,N}$ is the Weil pairing e_N, and $e_{r,N}$ is G_F-equivariant, bilinear, and alternating.

However, the situation is not so nice if $r > 1$. Fixing a generator ζ of $\boldsymbol{\mu}_N$, there is an isomorphism $h_{r,N} : \boldsymbol{\mu}_N^{\otimes r} \to \boldsymbol{\mu}_N$, induced by $h_{r,N}(\zeta^{a_1} \otimes \cdots \otimes \zeta^{a_r}) = \zeta^{a_1 \cdots a_r}$. By Lemma 7.2, the map $h_{r,N}$ is G_F-equivariant if and only if $q^{r-1}-1$ is divisible by N. Thus for fixed $r > 1$, the Galois-equivariance of the composition $h_{r,N} \circ e_{r,N}$ holds for only finitely many values of N. Therefore for $r > 1$, the maps $h_{r,\ell} \circ f_{r,\ell} : A[\ell]^{2r} \to \boldsymbol{\mu}_\ell$ do not in any meaningful sense fit into a "compatible system" of mod ℓ maps for infinitely many primes ℓ. Further, the isomorphism $h_{r,N}$ can only be computed by solving a Diffie-Hellman-like problem, namely, given $\zeta^{a_1},\ldots,\zeta^{a_r}$, find $\zeta^{a_1 \cdots a_r}$ (without knowing $a_1,\ldots,a_r$).

7.3. Alternating multilinear maps. We next consider alternating multilinear forms (the Weil pairing is one such). We show that for d-dimensional abelian varieties with $d > 1$, there are only finitely many primes ℓ for which a non-degenerate alternating multilinear form from $A[\ell]^{2d}$ to $\boldsymbol{\mu}_\ell$ can be Galois-equivariant. In other words, one only obtains a system of alternating, non-degenerate, Galois-equivariant multilinear forms from $A[\ell]^{2d}$ to $\boldsymbol{\mu}_\ell$ for infinitely many primes ℓ when the maps are bilinear pairings and A is an elliptic curve.

LEMMA 7.3. *Suppose V is an n-dimensional $\mathbb{F}_\ell$-vector space. Then:*
- (a) *there is a unique (up to scaling) multilinear alternating form $f : V^n \twoheadrightarrow \boldsymbol{\mu}_\ell$;*
- (b) *if $F = \mathbb{F}_q$, $\ell \nmid q$, and $\rho : G_F \to \mathrm{Aut}(V)$ is a homomorphism defining a Galois action on V, then f is G_F-equivariant if and only if $\chi_\ell = \det(\rho)$, where $\chi_\ell : G_F \to \mathbb{F}_\ell^\times$ is the cyclotomic character.*

PROOF. It is well-known that the set of alternating n-multilinear maps from an n-dimensional vector space to a one-dimensional vector space is one-dimensional. We thus have (a). Fix a generator ζ of $\boldsymbol{\mu}_\ell$ and a basis $\{v_1,\ldots,v_n\}$ of V over $\mathbb{F}_\ell$. Define

$$f(w_1,\ldots,w_n) = \zeta^{\det A}$$

with $w_i = \sum_{j=1}^n a_{ij}v_j$ for $i = 1,\ldots,n$ and $A = (a_{ij}) \in \mathrm{M}_n(\mathbb{F}_\ell)$. Then f is multilinear and alternating, and is the unique such map, up to the choice of generator and basis. Now f is G_F-equivariant if and only if for every $\sigma \in G_F$, we have $\sigma(f(v_1,\ldots,v_n)) = f(\sigma(v_1),\ldots,\sigma(v_n))$. Since $\sigma(f(v_1,\ldots,v_n)) = \sigma(\zeta) = \zeta^{\chi_\ell(\sigma)}$ and

$$f(\sigma(v_1),\ldots,\sigma(v_n)) = f(\rho(\sigma)(v_1),\ldots,\rho(\sigma)(v_n)) = \zeta^{\det \rho(\sigma)},$$

we have (b). □

PROPOSITION 7.4. *Suppose that ℓ is prime, that $F = \mathbb{F}_q$, that $\ell \nmid q$, that A is a d-dimensional abelian variety over F, and that $V = A[\ell]$ is the G_F-module of ℓ-torsion on A. Then the alternating multilinear form f defined in Lemma 7.3(a) above (with $n = 2d$) is G_F-equivariant if and only if $q^{d-1} - 1$ is divisible by ℓ.*

PROOF. Let $\rho : G_F \to \mathrm{Aut}(V)$ be the mod ℓ representation for A. Writing Φ for the Frobenius element of G_F, then $\det \rho(\Phi)$ is the constant term in the characteristic polynomial of Φ acting on $A[\ell]$, so $\det \rho(\Phi) = q^d = \chi_\ell(\Phi)^d$. Since Φ generates G_F, we have $\det \rho = \chi_\ell^d$. The result now follows from Lemmas 7.3(b) and 7.2. □

From the point of view of cryptography, one problem with the above alternating map f is that to compute it one must express elements of V in terms of the basis $\{v_1, \ldots, v_n\}$, and this amounts to solving the discrete log problem. For example, if $\{P_1, \ldots, P_{2d}\}$ is an $\mathbb{F}_\ell$-basis for $A[\ell]$, and $Q = nP_1$, then to compute $f(Q, Q_2, \ldots, Q_{2d})$ one begins by trying to compute n, which is the discrete log of Q with respect to P_1.

Another problem is that Proposition 7.4 provides evidence that when $d > 1$, this form f is not a very natural map, and therefore is not likely to be easily computable. In particular, for fixed A (and therefore q and d), these maps are Galois-equivariant for only finitely many primes ℓ. Though these maps are defined on an algebraic object, namely an abelian variety, they are not in general themselves algebraic, since they are not in general Galois-equivariant. We elaborate on this further in what follows.

7.4. Motives. If an n-multilinear map is computable, it is reasonable to expect it to come from geometry, as is the case for Weil and Tate pairings when $n = 2$. Failing that, one might hope that it at least comes from a motive.

From now on, we consider forms that are not necessarily alternating. In Corollary 7.7 we will show that if the desired n-multilinear map comes from a motive over a finite field, and is part of a system of Galois-equivariant mod ℓ maps for infinitely many primes ℓ, then $n = 1$ or 2 (and the motive has weight 2 or 1, respectively). For $n = 1$, the identity isomorphism $\boldsymbol{\mu}_\ell \to \boldsymbol{\mu}_\ell$ gives trivial 1-multilinear maps of weight 2 motives. Weil or Tate pairings on abelian varieties give rise to 2-multilinear maps of weight 1 motives. Note (Remark 2.7 of [**22**]) that the category of motives over finite fields is generated by Artin motives (which have weight 0) and abelian varieties (which have weight 1). Corollary 7.7 provides evidence that the desired forms will be motivic only in the case of bilinear pairings on abelian varieties and in trivial cases.

As alluded to in §7.2, it is not easy to tell when two elements of $\boldsymbol{\mu}_\ell^{\otimes n}$ are the same, if $n > 1$. For purposes of cryptography, we will therefore only consider the cases where the range is $\boldsymbol{\mu}_\ell$, $\mathbb{Z}/\ell\mathbb{Z}$, or $\mathrm{Hom}(\boldsymbol{\mu}_\ell, \mathbb{Z}/\ell\mathbb{Z})$ in what follows.

THEOREM 7.5. *Suppose that ℓ is prime, that $F = \mathbb{F}_q$, that $\ell \nmid q$, and that $V_1, \ldots, V_n$ are finite-dimensional $\mathbb{F}_\ell$-vector spaces with G_F-actions. Write Φ for the Frobenius element of G_F. Suppose $f : V_1 \times \cdots \times V_n \twoheadrightarrow \boldsymbol{\mu}_\ell$ is multilinear and G_F-equivariant. Then there are $\alpha_1, \ldots, \alpha_n \in \bar{\mathbb{F}}_\ell$ such that $\alpha_1 \cdots \alpha_n = q$ and for each i, α_i is an eigenvalue of Φ acting on V_i. If $\boldsymbol{\mu}_\ell$ is replaced by $\mathbb{Z}/\ell\mathbb{Z}$ (respectively, $\mathrm{Hom}(\boldsymbol{\mu}_\ell, \mathbb{Z}/\ell\mathbb{Z})$), then q is replaced by 1 (respectively, $1/q$) in the conclusion.*

Proof. The multilinear map f gives rise to a linear map $\tilde{f} : V_1 \otimes \cdots \otimes V_n \twoheadrightarrow \boldsymbol{\mu}_\ell$, as in Remark 7.1. Since f is G_F-equivariant, we have

$$\tilde{f} \circ \Phi^{\otimes n}(x_1 \otimes \cdots \otimes x_n) = \tilde{f}(\Phi(x_1) \otimes \cdots \otimes \Phi(x_n))$$

$$= \Phi \circ \tilde{f}(x_1 \otimes \cdots \otimes x_n) = \tilde{f}(x_1 \otimes \cdots \otimes x_n)^q.$$

Therefore, $\tilde{f}(\Phi^{\otimes n} - q)(x_1 \otimes \cdots \otimes x_n) = 0$ for all $x_i \in V_i$ and $1 \leq i \leq n$. Since $\tilde{f} \neq 0$, the map $\Phi^{\otimes n} - q$ does not surject onto $V_1 \otimes \cdots \otimes V_n$, and thus is not injective. Therefore, q is an eigenvalue of $\Phi^{\otimes n}$. By Proposition 11 on p. A.VII.39 of [**5**] and induction, the set of eigenvalues of $\Phi^{\otimes n}$ is

$$\{\alpha_1 \cdots \alpha_n : \quad \alpha_i \text{ is an eigenvalue for the action of } \Phi \text{ on } V_i\}.$$

Since the Galois action on $\mathbb{Z}/\ell\mathbb{Z}$ is trivial, q is replaced by 1 in the above, if $\boldsymbol{\mu}_\ell$ is replaced by $\mathbb{Z}/\ell\mathbb{Z}$. Similarly, q is replaced by $1/q$ if $\boldsymbol{\mu}_\ell$ is replaced by its dual. □

As a special case, note that if $F = \mathbb{F}_q$, $\ell \nmid q$, V is an n-dimensional $\mathbb{F}_\ell$-vector space with a G_F-action, $f : V^n \twoheadrightarrow \boldsymbol{\mu}_\ell$ is multilinear, alternating, and G_F-equivariant, and S is the set of eigenvalues of the action of Φ on V, then Lemma 7.3 shows that $\prod_{\alpha \in S} \alpha^{n_\alpha} = q$, where n_α is the multiplicity of α as an eigenvalue.

Corollary 7.6. *Suppose $M_1, \ldots, M_n$ are motives over $F = \mathbb{F}_q$ that are homogeneous of weights $m_1, \ldots, m_n$, respectively. Assume the Tate Conjecture holds for ζ-functions of smooth projective varieties over finite fields (see (1.14) of* [**22**]*). If ℓ is prime and $\ell \nmid q$, let $(M_i)_\ell$ be the mod ℓ realization of M_i. Suppose S is an infinite set of primes ℓ such that $\ell \nmid q$ and such that there is a G_F-equivariant multilinear homomorphism $f_\ell : (M_1)_\ell \times \cdots \times (M_n)_\ell \twoheadrightarrow \boldsymbol{\mu}_\ell$. Then $m_1 + \cdots + m_n = 2$. If $\boldsymbol{\mu}_\ell$ is replaced by $\mathbb{Z}/\ell\mathbb{Z}$ (respectively,* $\mathrm{Hom}(\boldsymbol{\mu}_\ell, \mathbb{Z}/\ell\mathbb{Z})$*), then $m_1 + \cdots + m_n = 0$ (respectively, -2).*

Proof. Write $S_i \subset \bar{\mathbb{Q}}$ for the set of eigenvalues of Frobenius acting on M_i. Let T be the finite set

$$T = \{q - \alpha_1 \cdots \alpha_n : \quad \alpha_i \in S_i\} \subset \bar{\mathbb{Q}}.$$

By Proposition 2.2 of [**22**], $|\alpha_i| = q^{m_i/2}$ (this follows from the Weil Conjectures, proved by Deligne in [**7**]). Thus, $|\alpha_1\alpha_2 \cdots \alpha_n| = q^{(m_1+\cdots+m_n)/2}$. Suppose $m_1 + \cdots + m_n \neq 2$. Then $0 \notin T$. Therefore there are only finitely many prime ideals of the ring $\bar{\mathbb{Z}}$ of algebraic integers that divide elements of the finite set T. However, by Theorem 7.5, for every $\ell \in S$ there is a prime ideal of $\bar{\mathbb{Z}}$ above ℓ that divides some element of T. Thus, S is finite. To finish the proof, replace q by 1 (resp., $1/q$) in the definition of T. □

Corollary 7.7. *Suppose M is a motive over $F = \mathbb{F}_q$ that is homogeneous of weight m. Assume the Tate Conjecture holds for ζ-functions of smooth projective varieties over finite fields (see (1.14) of* [**22**]*). If ℓ is prime and $\ell \nmid q$, let M_ℓ be the mod ℓ realization of M. Suppose S is an infinite set of primes ℓ such that $\ell \nmid q$ and such that there is a G_F-equivariant multilinear homomorphism $f_\ell : M_\ell^n \twoheadrightarrow \boldsymbol{\mu}_\ell$. Then $(m, n) = (2, 1)$ or $(1, 2)$. If $\boldsymbol{\mu}_\ell$ is replaced by $\mathbb{Z}/\ell\mathbb{Z}$, then $m = 0$. If $\boldsymbol{\mu}_\ell$ is replaced by* $\mathrm{Hom}(\boldsymbol{\mu}_\ell, \mathbb{Z}/\ell\mathbb{Z})$*, then $(m, n) = (-2, 1)$ or $(-1, 2)$.*

7.5. Tate pairings. We end with a brief discussion of Tate pairings (see Section 3.3 of [**9**] for more information). Suppose that $F = \mathbb{F}_q$, that $K = \mathbb{F}_{q^m}$, that ℓ is a prime divisor of $q^m - 1$, and that J is the Jacobian of a curve of genus ≥ 1 defined over F. Then (see Corollary 3.21 of [**9**]) the Tate (or Tate-Lichtenbaum) pairing induces a non-degenerate $\mathrm{Gal}(K/F)$-equivariant bilinear pairing

$$\langle\ ,\ \rangle : J(K)[\ell] \times J(K)[\ell] \to \boldsymbol{\mu}_\ell \subset K.$$

If $P \in J(K)[\ell]$ and G_1 is the group generated by P, then this pairing induces a $\mathrm{Gal}(K/F)$-equivariant pairing

$$\langle\ ,\ \rangle : G_1 \times G_1 \to \boldsymbol{\mu}_\ell.$$

If J is supersingular, then K is a "small" extension of F (see [**27**]), so this map is efficiently computable. Note that the Weil pairing is invariant under field extension, while the Tate pairing is not — changing the field K changes the map. Therefore in at least one way, the Tate pairing can be viewed as a less natural map. This augers well for the idea that useful pairings in cryptography could come from geometry, and yet not have all possible seemingly good properties (such as Galois-equivariance). We therefore conclude on the optimistic note that interesting geometric objects could still lead to useful n-multilinear maps with $n > 2$.

8. Conclusions

We gave strong motivation for constructing cryptographic n-multilinear maps. We showed that such maps give low-bandwidth broadcast encryption schemes, unique signature schemes, verifiable pseudo random functions, and a one-round conference key exchange protocol. We hope this ample motivation will eventually lead to an efficient construction for a cryptographic multilinear map. We also give evidence that such maps might have to either come from outside the realm of algebraic geometry, or occur as "unnatural" computable maps arising from geometry.

Acknowledgments

The authors would like to thank Karl Rubin, Bjorn Poonen, and Joe Buhler for helpful discussions.

References

[1] M. Bellare, P. Rogaway, *Random oracles are practical: a paradigm for designing efficient protocols*, in ACM Conference on Computers and Communication Security (1993), 62–73.

[2] D. Boneh, *The decision Diffie-Hellman problem*, in Proc. ANTS III, Lecture Notes in Computer Science **1423** (1998), Springer, 48–63.

[3] D. Boneh, M. Franklin, *Identity based encryption from the Weil pairing*, in Proc. Crypto 2001, Lecture Notes in Computer Science **2139** (2001), Springer, 213–229.

[4] D. Boneh, B. Lynn, and H. Shacham, *Short signatures from the Weil pairing*, Proc. Asiacrypt 2001, Lecture Notes in Computer Science **2248** (2001), Springer, 514–532.

[5] N. Bourbaki, Elements of mathematics, Algebra II, Chapters 4–7, Springer, 1990.

[6] G. C. Chick, S. E. Tavares, *Flexible access control with master keys*, in Proc. Crypto 1989, Lecture Notes in Computer Science **435** (1990), Springer, 316–322.

[7] P. Deligne, *La conjecture de Weil. I*, Inst. Hautes Études Sci. Publ. Math. **43** (1974), 273–307.

[8] A. Fiat and M. Naor, *Broadcast encryption*, in Proc. Crypto 1993, Lecture Notes in Computer Science **773** (1993), Springer, 480–491.

[9] G. Frey, *Applications of arithmetical geometry to cryptographic constructions*, in Finite fields and applications (Augsburg, 1999) (2001), Springer, 128–161.

[10] G. Frey, H-G. Rück, *A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves*, Math. Comp. **62** (1994), 865–874.

[11] S. Goldwasser, S. Micali, R. Rivest, *A digital signature scheme secure against adaptive chosen message attacks*, SIAM J. of Computing, **17**, No. 2 (1988), 281-308.

[12] S. Goldwasser, R. Ostrovsky, *Invariant signatures and non-interactive zero knowledge proofs are equivalent*, in Proc. Crypto 1992, Lecture Notes in Computer Science **740** (1992), Springer, 228–244.

[13] D. Halevi, A. Shamir, *The LSD broadcast encryption scheme*, in Proc. Crypto 2002, Lecture Notes in Computer Science **2442** (2002), Springer, 47–60.

[14] U. Jannsen, S. Kleiman, J-P. Serre (eds.), Motives, Proc. Symp. Pure Math., vol. 55, Amer. Math. Soc., 1994, part 1.

[15] A. Joux, *A one round protocol for tripartite Diffie-Hellman*, Proc. ANTS IV, Lecture Notes in Computer Science **1838** (2000), 385–394.

[16] A. Joux, K. Nguyen, *Separating Decision Diffie-Hellman from Diffie-Hellman in cryptographic groups*, available from `eprint.iacr.org`.

[17] A. Lysyanskaya, *Unique signatures and verifiable random functions from DH-DDH separation*, in Proc. Crypto 2002, Lecture Notes in Computer Science **2442** (2002), Springer, 597–612.

[18] A. Menezes, Elliptic curve public key cryptosystems, Kluwer Academic Publishers, Boston, MA, 1993.

[19] A. Menezes, T. Okamoto, S. Vanstone, *Reducing elliptic curve logarithms to logarithms in a finite field*, IEEE Transactions on Information Theory **39** (1993), 1639–1646.

[20] S. Micali, M. O. Rabin, S. P. Vadhan, *Verifiable random functions*, in Proc. 40th IEEE Symposium on Foundations of Computer Science (FOCS) (1999), 120–130.

[21] V. Miller, *Short programs for functions on curves*, unpublished manuscript.

[22] J. Milne, *Motives over finite fields*, in [**14**], 401–459.

[23] S. Mitsunari, R. Sakai, M. Kasahara, *A new traitor tracing*, IEICE Trans. Fundamentals **E85-A**, no. 2 (2002), 481–484.

[24] D. Naor, M. Naor, J. Lotspiech, *Revocation and Tracing Schemes for Stateless Receivers*, in Proc. Crypto 2001, Lecture Notes in Computer Science **2139** (2001), 41–62.

[25] M. Naor, O. Reingold, *Number-theoretic constructions for efficient pseudo random functions*, in Proc. 38th IEEE Symposium on Foundations of Computer Science (FOCS) (1997), 458–467.

[26] S. Pohlig, M. Hellman, *An improved algorithm for computing discrete logarithms over $GF(p)$ and its cryptographic significance*, IEEE Transactions on Information Theory **24** (1978), 106–110.

[27] K. Rubin, A. Silverberg, *Supersingular abelian varieties in cryptology*, in Proc. Crypto 2002, Lecture Notes in Computer Science **2442** (2002), Springer, 336–353.

[28] R. Sakai, K. Ohgishi, M. Kasahara, *Cryptosystems based on pairing*, SCIC 2000-C20, Okinawa, Japan, January 2000.

[29] A. J. Scholl, *Classical motives*, in [**14**], 401–459.

[30] M. Steiner, G. Tsudik, M. Waidner, *Diffie-Hellman key distribution extended to group communication*, in Proc. 3rd ACM Conference on Communications Security (1996), 31–37.

Department of Computer Science, Stanford University, Stanford, CA, 94305, USA
E-mail address: `dabo@cs.stanford.edu`

Department of Mathematics, Ohio State University, Columbus, OH 43210, USA
E-mail address: `silver@math.ohio-state.edu`