A Proposed KDM-Secure Scheme to Evaluate k-DNF Formulas

Zachary Stier

Junior Paper, Spring 2019

This paper is submitted to the Princeton University Department of Mathematics in partial fulfillment of junior year Independent Work.

Advisor: Professor Mark Zhandry
Second reader: Professor Zeev Dvir

I pledge my honor that this paper represents my own work in accordance with University regulations.

Abstract

Boneh, Halevi, Hamburg, and Ostrovsky present in [4] an ElGamal-like encryption scheme relying on the decisional Diffie-Hellman problem in which encryptions of secret keys may be published without fear of undermining the security of a public key-secret key pair—a security notion known as circular security, a special case of KDM-security. Boneh, Goh and Nissim present in [3] a different ElGamal-like semantically secure encryption scheme in which products of ciphertexts faithfully correspond to encryptions of the products of the plaintexts, but is only equipped to handle 2-DNF formulas. As a capstone of work from this term, this paper concerns both of these results in greater generality, extending the former to rely on the k-linear assumption of [1, 8, 11] and discusses the latter in the context of k-linear maps, enabling the evaluation of k-DNF formulas, before describing a new proposed encryption scheme relying on the k-linear assumption that is KDM-secure and allows for the evaluation of k-DNF formulas given a k-linear map.

Contents

Notation

- $\mathbf{N}$ denotes the natural numbers $\{1, 2, \dots\}$, $\mathbf{Z}$ denotes the integers, $\mathbf{Z}_n$ denotes the integers modulo n, and $\mathbf{R}$ denotes the real numbers.
- $\mathbf{P}[A]$ denotes the probability of event A. $\mathbf{E}[X]$ denotes the expected value of random variable X.
- For $n \in \mathbf{N}$, $[n]$ denotes the subset of positive integers $\{1, \dots, n\}$.
- Vectors are typically denoted by boldface characters, and the vector $\mathbf{a}$ has entries $(a_1, a_2, \dots)$.
- $\mathbf{e}_i$ are the standard basis vectors, i.e. those with 1 in position i and 0 elsewhere.
- $a \xleftarrow{R} S$ indicates that a is selected uniformly at random from the finite set S.
- $A \subset_n B$ means that $A \subset B$ and $|A| = n$.
- $\exp_a b$ denotes a^b.
- $\log_b : \mathbf{Z}_p \to \mathbf{N}$ denotes the least positive integer solving the base-b discrete logarithm problem.
- $\log : \mathbf{G} \to \{0, 1\}$ denotes $\log x = \begin{cases} 0 & x = 1 \\ 1 & \text{else} \end{cases}$.

- $\operatorname{rank}_n \mathbf{G}^{a\times b}$ denotes the set of rank-n $a \times b$ matrices with elements in $\mathbf{G}$, while $\operatorname{rank} M$ denotes the rank of the matrix M.
- $(A|B)$ denotes the columns of matrix A followed by the columns of B.
- $\mathbf{v} \times \mathbf{w}$ denotes the $m \times n$ matrix $(v_i w_j)_{i\in[m],j\in[n]}$, for $\mathbf{v} \in \mathbf{Z}_{|\mathbf{G}|}^m$ and $\mathbf{w} \in \mathbf{G}^n$.

1 Introduction

Many public-key encryption schemes do not admit obvious weaknesses when their secret key is encrypted, though it is certainly easy to contrive those that *do* become immediately weakened, for instance as noted in [4] by ensuring that a secret key always encrypts to itself. Practically speaking, it is not unreasonable to expect or hope for a drive containing its own password to still be securely encryptable. It is therefore advantageous to construct a scheme around this type of security. More powerfully, we are interested in a cryptosystem (Enc, Dec) for which, given n public key-secret key pairs (pk_i, sk_i), security is intact even if $\text{Enc}(pk_i, sk_j)$ is published for $1 \leqslant i, j \leqslant n$. This is the notion of KDM-security, a generalized case of circular security, in which only $\text{Enc}(pk_i, sk_{i+1})$ is published.

Perhaps similarly, often public-key encryption schemes are homomorphic in a single operation; e.g. ElGamal is multiplicatively homomorphic, as the componentwise product of encryptions $(g^{r_1}, m_1 g^{r_1 x})$ and $(g^{r_2}, m_2 g^{r_2 x})$ corresponds to

$$\left(g^{r_1+r_2}, (m_1 m_2) g^{(r_1+r_2)x}\right).$$

However, it is much more difficult to incorporate homomorphism in a second operation. This feat was first accomplished in full by Gentry in [6], using matrices and lattices and basing security on the learning with errors problem, and in this regime has been improved in the ensuing decade. However, it was also been attacked earlier in the form of an ElGamal-like scheme, though only permitting a bounded number of multiplications; this is the route that we pursue here.

In this paper, I will discuss encryption schemes that satisfy each of these objectives, followed by a proposal for a new encryption scheme that is both circularly secure and can evaluate k-DNF formulas. This scheme primarily relies on the security notions and proofs in [4] but also crucially relies on homomorphism and decryption ideas from [3]. One property of note is that the k-DNF evaluation property is ported from a semiprime-order group cryptosystem to a prime-order group in this new scheme.

This paper represents a cumulation of the content that I have learned and thought about over the course of this semester. Before discussing some of the existing literature, I will review relevant ideas of hardness and security in §2.1 and §2.2, respectively. Pollard's lambda algorithm for discrete logarithms (invoked in [3]) is discussed in §2.3, and the mathematical ideas underlying elliptic curves and the Weil pairing, used in the $k = 2$ case of §4, are reviewed in §2.4. Then, while presenting the results of [4] in §3, instead of relying on the decisional Diffie-Hellman assumption, we will instead see the results modified to fit the k-linearity assumption. (This is a proof briefly mentioned in [4] but that has been carried out fully here.) Doing so makes the security proof in §5 more immediate. We will treat the scheme in [3], presented in §4, in slightly greater generality—as having the capacity to evaluate k-DNF formulas, rather than 2-DNF formulas. §5 contains a full discussion of the proposed scheme.

2 Cryptographic, algorithmic, and mathematical preliminaries

2.1 Hardness assumptions

2.1.1 The k-linearity and decisional Diffie-Hellman assumptions

For an additive group $\mathbf{G}$ of order p, define:

- $\mathcal{P}_{k\text{-Lin}}$ is the set of $(2k+2)$-tuples of the form $\left(g_1, r_1 g_1, \ldots, g_k, r_k g_k, h, \sum_{i=1}^{k} r_i h\right)$ for $g_1, \ldots, g_k, h \in \mathbf{G}$ and $r_1, \ldots, r_k \in \mathbf{Z}_p$.
- $\mathcal{R}_{k\text{-Lin}}$ is the set of $(2k+2)$-tuples of the form $(g_1, r_1 g_1, \ldots, g_k, r_k g_k, h, \sigma h)$ for $g_1, \ldots, g_k, h \in \mathbf{G}$ and $r_1, \ldots, r_k, \sigma \in \mathbf{Z}_p$.

The k-linearity security game consists of a challenger $\mathcal{D}$ that draws $b \xleftarrow{R} \{0,1\}$ and gives adversary $\mathcal{A}$ the $(2k+2)$-tuple t for

$$\begin{cases} t \xleftarrow{R} \mathcal{P}_{k\text{-Lin}} & b = 0 \\ t \xleftarrow{R} \mathcal{R}_{k\text{-Lin}} & b = 1 \end{cases}.$$

$\mathcal{A}$ must ascertain the value of b. The advantage of $\mathcal{A}$ in this game is defined as

$$k\text{-LinAdv}[\mathcal{A}, \mathbf{G}] := \left| \mathbf{P}[\mathcal{A} \text{ correctly guesses } b] - \frac{1}{2} \right|.$$

(For notational convenience, we say that the k-linearity challenge arrives as the $(2k+2)$-tuple $(\alpha_1, \beta_1, \ldots, \alpha_k, \beta_k, \alpha_{k+1}, \beta_{k+1})$.)

The **k-linearity assumption** asserts that there does not exist an adversary $\mathcal{A}$ with computational power polynomial in p with non-negligible advantage in p. The k-linearity assumption (under various similar names) was discussed in [8, 11] with both drawing from [1].

We note that the hardness of 1-linearity is equivalent to the **decisional Diffie-Hellman assumption** (DDH), which is usually formulated (over multiplicative $\mathbf{G}$ as asking an adversary to distinguish between a 4-tuple of the form $(g, g^a, g^b, g^{ab}) \leftarrow \mathcal{P}_{\text{DDH}}$ and $(g, g^a, g^b, g^c) \leftarrow \mathcal{R}_{\text{DDH}}$; instead, in the language of k-linearity, h is "shorthand" for g^b, rendering the tuples as $(g, g^a, h, h^a) \leftarrow \mathcal{P}_{1\text{-Lin}}$ and $(g, g^a, h, h^c) \leftarrow \mathcal{R}_{1\text{-Lin}}$.

We are interested here in k-linearity because when operating in a group equipped with an efficiently-computable k-linear form, for instance that of [3] having the modified Weil pairing $\widehat{e}$ with $k = 2$, it is clear that DDH is trivial by considering, for input $(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$, $\widehat{e}(\alpha_1, \alpha_4) \stackrel{?}{=} \widehat{e}(\alpha_2, \alpha_3)$.

2.1.2 The subgroup decision assumption

The **subgroup decision assumption**, introduced in [3], asserts that it is difficult to discern whether an arbitrary element of a semiprime group has prime order. The assumption applies to algorithms $\mathcal{G}$ for generating groups of semiprime order with bilinear maps. Put precisely: let $\mathcal{G}$ take as its argument the security parameter $\lambda \in \mathbf{N}$. Generate λ-bit primes q_1, q_2, a group $\mathbf{G}$ of order $n := q_1 q_2$ with generator

g, another group $\mathbf{G}'$ also of order n, and a bilinear map $e : \mathbf{G}^2 \to \mathbf{G}'$. $\mathcal{G}$ outputs the tuple $(q_1, q_2, \mathbf{G}, \mathbf{G}', e)$. The subgroup decision game consists of a challenger $\mathcal{D}$ that runs $\mathcal{G}(\lambda)$ and draws $b \xleftarrow{R} \{0,1\}$ and $x \xleftarrow{R} \mathbf{G}$, giving $(n, \mathbf{G}, \mathbf{G}', e, x^{b(q_2-1)+1})$ to adversary $\mathcal{A}$. $\mathcal{A}$ must ascertain the value of b. The advantage of $\mathcal{A}$ in this game is defined as

$$\mathrm{SDAdv}[\mathcal{A}, \mathcal{G}](\lambda) := \left| \mathbf{P}[\mathcal{A} \text{ guesses correctly}] - \frac{1}{2} \right|.$$

The subgroup decision assumption is that there does not exist an adversary $\mathcal{A}$ running in polynomial time in λ with $\mathrm{SDAdv}[\mathcal{A}, \mathcal{G}](\lambda)$ non-negligible in λ.

2.2 Security notions

In this section, we will introduce the types of security satisfied by the schemes described in §3–§5.

2.2.1 Semantic security

The **semantic security game** for the encryption scheme $\mathcal{E} = (\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ consists of the following actions: a challenger $\mathcal{D}$ runs $\mathrm{Gen}(\lambda)$, returning a message space $\mathcal{M}$ (deterministically), a public key pk, and a secret key. $\mathcal{M}$ and pk are given to an adversary $\mathcal{A}$ which is allowed running time polynomial in λ. $\mathcal{A}$ produces $m_0, m_1 \in \mathcal{M}$ for $\mathcal{D}$, which draws $b \xleftarrow{R} \{0,1\}$ and returns to $\mathcal{A}$ an encryption $\mathrm{Enc}(pk, m_b)$. $\mathcal{A}$ must ascertain the value of b. The advantage of $\mathcal{A}$ in this game is defined as

$$\mathrm{SemAdv}[\mathcal{A}, \mathcal{E}](\lambda) := \left| \mathbf{P}[\mathcal{A} \text{ correctly guesses } b] - \frac{1}{2} \right|.$$

$\mathcal{E}$ is **semantically secure** if $\mathrm{SemAdv}[\mathcal{A}, \mathcal{E}](\lambda)$ is negligible in λ for all $\mathcal{A}$ running in polynomial time in λ.

2.2.2 KDM-security and circular security

A scheme $\mathcal{E} = (\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$, with $\mathrm{Gen}(\cdot)$ returning a message space $\mathcal{M}$ (deterministically), a public key, and a secret key lying in S (the latter two nondeterministically), has **key-dependent message security** (KDM-security) if not only are encryptions of chosen plaintexts indistinguishable from encryptions of a fixed message (e.g. a string of all 0's), but also an adversary can query about a function of the secret keys (of course, without being told the keys). That is to say, given $\mathcal{C}$ a set of functions $f : S^n \to \mathcal{M}$, the KDM-security game consists of a challenger $\mathcal{D}$ that draws $b \xleftarrow{R} \{0,1\}$ and gives adversary $\mathcal{A}$ the message $m \in \mathcal{M}$ as well as the n pairs $(pk_1, sk_1), \ldots, (pk_n, sk_n)$ arising from n calls of $\mathrm{Gen}(\lambda)$. Let $\mathbf{s}$ be the length-$n\ell$ concatenation of the secret keys (where each secret key has fixed length ℓ). $\mathcal{A}$ may send $\mathcal{D}$ a number polynomial in λ of pairs (i, f) for $1 \leqslant i \leqslant n$ and $f \in \mathcal{C}$, to which $\mathcal{D}$ replies with

$$\begin{cases} \mathrm{Enc}(pk_i, f(\mathbf{s})) & b = 0 \\ \mathrm{Enc}(pk_i, m) & b = 1 \end{cases}$$

$\mathcal{A}$ must ascertain the value of b. The advantage of $\mathcal{A}$ in this game is defined as

$$\mathrm{KDMAdv}_{\mathcal{C}}^{(n)}[\mathcal{A}, \mathcal{E}](\lambda) := \left| \mathbf{P}[\mathcal{A} \text{ correctly guesses } b] - \frac{1}{2} \right|.$$

$\mathcal{E}$ is n-**way KDM-secure with respect to** $\mathcal{C}$ if $\mathrm{KDMAdv}_{\mathcal{C}}^{(n)}[\mathcal{A}, \mathcal{E}](\lambda)$ is negligible in λ for all $\mathcal{A}$ running in polynomial time in λ.

$\mathcal{E}$ has **clique security** if it is n-way KDM-secure with respect to $\mathcal{C}$ where $\mathcal{C}$ contains the n "selector functions" $f_i : \mathbf{s} \mapsto sk_i$ and the $|\mathcal{M}|$ constant functions. **Circular security** is a weakened form of clique security in which the pairs (i, f_j) may be queried to the challenger only if $i - j \equiv 1 \pmod{n}$; we shall only be concerned here with security notions that are at least as strong as clique security, and hence strictly stronger than circular security. Indeed, the clique case shall be covered by considering $\mathcal{C}_N$ to be the set of affine functions $\mathbf{Z}_p^N \to \mathbf{Z}_p^N$ characterized by $\mathbf{x} \mapsto A\mathbf{x} + \mathbf{b}$ for A a matrix and $\mathbf{b}$ a vector.

2.3 Pollard's lambda algorithm

The following method is due to Pollard in [10]. In its most general form, the algorithm computes intersections between pseudorandom subsets of a large finite group. (In terms of kangaroos, the tame kangaroo sets traps for the wild kangaroo; both leap in a pseudorandom fashion.) What we shall see now, however, is the special case of the algorithm, used to compute a discrete logarithm with high probability.

It is also known under the name of **Pollard's kangaroo algorithm** because of the analogy that Pollard draws in his paper to using a tame kangaroo to catch a wild kangaroo. The notion of a "lambda" arises from viewing one of the pseudorandom variables as having a straight-line path that collides with the middle of the other's straight-line path.

2.3.1 The algorithm

We begin with a prime p, a base $r \in \mathbf{Z}$, a value $q \in \mathbf{Z}_p$, integers $A < B$, and a guarantee that there exists $e \in [A, B] \cap \mathbf{Z}$ such that $r^e \equiv q \pmod{p}$. We wish to compute $e = \log_r q$.

First, select a tuning parameter θ (see the next section for discussion on θ's value). Next, select a finite set $S \subset \mathbf{N}$ of size $|S| \ll \sqrt{B - A}$ and construct a pseudorandom function $f : \mathbf{Z}_p \to S$ (i.e. a deterministic function, computable in polynomial time, that is computationally indistinguishable from truly random). We precompute $r^s \pmod{p}$ for each $s \in S$. (This is where we need the requirement $|S| \ll \sqrt{B - A}$—otherwise this step is completely impractical.) Select $N \in \mathbf{N}$ such that $N \approx \theta m$ where m is the average value in S. We now construct the sequence $\{x_i\}_{i=0}^N$ as follows: $x_0 := r^B \pmod{p}$ and $x_{i+1} := x_i r^{f(x_i)} \pmod{p}$. Define $d_i := \sum_{j=0}^{i-1} f(x_j)$. We now compute a new sequence $\{x'_i\}_{i \in \mathbf{N}}$ as follows: $x'_0 := q$ and $x'_{i+1} := x'_i r^{f(x'_i)} \pmod{p}$. Define $d'_i := \sum_{j=0}^{i-1} f(x'_j)$. If there exists $M \in \mathbf{N}$ such that $x'_M = x_N$, then we return

$$B + d_N - d'_M.$$

Otherwise, once $d'_M > d_N + B - A$, halt, select new values of S and f, and start over.

2.3.2 Analysis: correctness, timing, and likelihood of success

We shall first show that it is correct to halt when $x'_M = x_N$. Noting that d_i satisfies $x_i = x_0 r^{d_i}$ and d'_i satisfies $x'_i = x'_0 r^{d'_i}$, we expand and have $q r^{d'_M} = r^B r^{d_N}$ and so $\log_r q = B + d_N - d'_M$.

If $d'_M > d_N + B - A$, we know that our choice of S and f will be insufficient, as d'_M is monotonically increasing, and we would have $B + d_N - d'_M < A$, a violation of $\log_r q \in [A, B]$. So, if an arbitrary choice of S and f has high probability of success and a full iteration for a given choice runs sufficiently quickly, then this algorithm is viable.

We compute timing and likelihood of success in terms of θ.

The number of operations is on the order of $N + M$, so we with to compute $\mathbf{E}[N+M] = N + \left(\frac{1}{2}\frac{B-A}{m} + N\right)$ because in expectation we halt about halfway through the algorithm. Temporarily using $m = \alpha\sqrt{B-A}$ for some $\alpha \in \mathbf{R}_{>0}$, this expectation is $\sqrt{B-A}\left(2\alpha\theta + \frac{1}{2\alpha}\right)$, taking its minimum of $2\sqrt{\theta(B-A)}$ at $\alpha = \frac{1}{2\sqrt{\theta}}$ so we wish to have $m \approx \frac{1}{2}\sqrt{\frac{B-A}{\theta}}$.

At each step, the likelihood of halting is $\frac{1}{m}$, so the likelihood of failure in the N attempts, by (pseudo)randomness, is approximated by $\left(1 - \frac{1}{m}\right)^N$ and thus success occurs with likelihood $1 - \left(1 - \frac{1}{m}\right)^N = 1 - \left(1 - \frac{1}{m}\right)^{m\theta} \approx 1 - \frac{1}{e^\theta} > 0.98$ for $\theta \geqslant 4$. Therefore the expected number of times the whole algorithm must be run is $\frac{1}{1-\frac{1}{e^\theta}} = 1 + \frac{1}{e^\theta - 1} < 1.02$ for $\theta \geqslant 4$.

2.4 Elliptic curves

In this section, we will state some basic elliptic curve facts and prove some elementary results (which can be found, e.g., in [12]). We will then see a bilinear function known as the Weil pairing. (That portion largely comes from [2].) Because this section is so heavy on definitions, we shall present it in a quantized definition-lemma format. Further, in the interest of brevity—since elliptic curves will not play a central role in the following cryptographic discourse, but are central to the scheme of [3]—we shall sketch or omit many of the more involved proofs.

Definition 1 (nonsingularity)**.** A plane curve given by $f(x, y) = 0$ is *nonsingular* if, for each point P on the curve,

$$\left(\frac{\partial}{\partial x} f(P), \frac{\partial}{\partial y} f(P)\right) \neq 0.$$

Definition 2 (elliptic curve)**.** An *elliptic curve* is a nonsingular plane cubic in Weierstrass form:

$$E : y^2 + a_1 xy + a_3 y = x^3 + a_2 x^2 + a_4 x + a_6.$$

Definition 3. $E(K)$ is the set of solutions to elliptic curve E over the field K along with a "point at infinity," $\mathcal{O}$.

$\mathcal{O}$ arises as $[0, 1, 0]$ from viewing E as a projective curve, in homogeneous coordinates X, Y, Z:

$$Y^2 Z + a_1 XYZ + a_3 YZ^2 = X^3 + a_2 X^2 Z + a_4 XZ^2 + a_6 Z^3.$$

Definition 4. For $P, Q \in E(K)$, let $P * Q$ be the third intersection point of the line connecting P and Q in $K^2 \cup \mathcal{O}$. Let $P + Q = \mathcal{O} * P * Q$.

Proposition 5. *$P * Q$ always exists (although it might need to be defined in terms of tangencies), $*$ is associative, and $(E(K), +)$ is an abelian group with identity $\mathcal{O}$.*

Proof. This is a classical but involved result, particularly for associativity; see e.g. Proposition 2.2 of [12]. ■

Definition 6 (torsion subgroup)**.** $E(K)[n]$ is the nth *torsion subgroup*, i.e. all elements of $E(K)$ of order n.

Lemma 7. *For $E : y^2 = x^3 + 1$, $\#E[\mathbf{F}_p] = p + 1$ for all primes $p > 3$ with $3 \nmid p - 1$.*

Proof. 3 is coprime to $\#\mathbf{F}_p^\times$, so cubing is injective, while squaring is 2-to-1. If $y^2 = 0$ then there is a unique x with $x^3 = -1$. Otherwise, for $z \in (\mathbf{F}_p^\times)^2$, there is a unique x with $x^3 = z - 1$, while there are two distinct y with $y^2 = z$. $\#(\mathbf{F}_p^\times)^2 = \frac{1}{2}(p-1)$. Adding in $\mathcal{O}$, the group has $1 + 1 + 2 \cdot \frac{1}{2}(p-1) = p + 1$ elements. ■

Definition 8 (divisor)**.** A *divisor* is a formal sum

$$\sum_{P \in E(\mathbf{F}_{p^2})} a_P(P),$$

where $a_P \in \mathbf{Z}$ and (P) distinguishes P as a generator of a free $\mathbf{Z}$-module. For a bivariate rational function $f : E(\mathbf{F}_{p^2}) \to \mathbf{R}$, its *divisor* is defined as

$$(f) = \sum_{P \in E(\mathbf{F}_{p^2})} \operatorname{ord}_f P \cdot (P),$$

where ord_f is the order of a point $P = (x_0, y_0)$ on the curve as a root of $f(x, y)$.

For instance, a line through two non-inverse points has three roots and a triple pole at $\mathcal{O}$, so its divisor looks like $(P) + (Q) + (R) - 3(\mathcal{O})$.

The index of this sum will be omitted going forward for ease of reading.

Definition 9 (principal divisor)**.** A *principal divisor* is a divisor $\mathcal{D}$ arising as (f) for some rational function f.

Lemma 10. *$\sum a_P(P)$ is a principal divisor iff $\sum a_P = 0$ and $\sum a_P P = \mathcal{O}$.*

Note that this last sum is of points as group elements.

Proof idea. $\to$ For f with $(f) = \sum a_P(P)$, we count the mulitplicities and note that they balance with the multiplicity of $\mathcal{O}$ once we view f projectively (to account for having $\mathcal{O}$ as a root).

$\leftarrow$ We construct f with $(f) = \sum a_P(P)$ such that its roots lie precisely at the points P with multiplicities a_P.

■

Definition 11. Define f evaluated at $\mathcal{D} = \sum a_P(P)$ as $f(\mathcal{D}) := \prod f(P)^{a_P}$.

We note that this is invariant under scaling f by a constant in $c \in \mathbf{F}_{p^2}$ since this produces a factor $\prod c^{a_P} = 1$.

Lemma 12 (Weil reciprocity)**.** *$f((g)) = g((f))$ for f and g with disjoint supports.*

Definition 13 (Weil pairing)**.** Fix n. For $P, Q \in E(\mathbf{F}_{p^2})[n]$ (abbreviated to $E[n]$), define $\mathcal{D}_P := n(P) - n(\mathcal{O})$ and $\mathcal{D}_Q := n(Q) - n(\mathcal{O})$. Let f_P and f_Q have $(f_P) = \mathcal{D}_P$ and $(f_Q) = \mathcal{D}_Q$. The *Weil pairing* $e : \mathbf{F}_{p^2}^2 \to \mathbf{R}$ is given by

$$e(P, Q) := \frac{f_P(\mathcal{D}_Q)}{f_Q(\mathcal{D}_P)}.$$

Lemma 14. *The Weil pairing has the following properties:*

(1) It is well-defined.

(2) e is bilinear in both components.

(3) e's range is actually μ_n, the nth roots of unity.

Proof idea. (1) Using alternate choices $\mathcal{D}_P$ and f_P, we note that they differ from the original by (g) and $\cdot g$ for some g; using Lemma 12, extraneous terms cancel.

(2) Because e is antisymmetric ($e(P, Q) = e(Q, P)^{-1}$ by inspection), it suffices to show for the first component. Using the evaluation of f_Q at $n(P + P') - n(\mathcal{O})$ and $f_{P+P'}$ at $\mathcal{D}_Q$, and cancelling terms arising from possibly new mutual zeroes/poles, we obtain the desired product.

(3) This follows by bilinearity e.g. in the first component (since the domain is in the n-torsion).

■

Because of (3), we may equivalently view e as having codomain any cyclic group of order n as we see fit.

Definition 15 (modified Weil pairing)**.** Let ζ be a nontrivial cubic root of unity in $\mathbf{F}_{p^2}$. Define $\phi : \mathbf{F}_{p^2}^2 \to \mathbf{F}_{p^2}^2$ by $(x, y) \mapsto (\zeta x, y)$. The *modified Weil pairing* $\widehat{e} : \mathbf{F}_{p^2}^2 \to \mu_n$ is given by

$$\widehat{e}(P, Q) = e(P, \phi(Q)).$$

Lemma 16. *The modified Weil pairing has the following properties:*

(1) ζ exists and ϕ is an automorphism on $E(\mathbf{F}_{p^2})$.

(2) If P generates a subgroup of order n then $\widehat{e}(P, P)$ also generates such a subgroup (in the image group).

Proof idea. (1) ζ exists because $3 \mid p^2 - 1$, so cubing is 3-to-1 on $\mathbf{F}_{p^2}$. Multiplication by scalar ζ is an injection; since there are no linear or quadratic x terms in the specification of E, $(\zeta x)^3 = x^3$ and hence ϕ sends points on E to points on E whose sum respects the group law.

(2) If not, then bilinearity would force P to generate a group of order (properly) dividing n, a contradiction.

■

Further, it is clear that $\widehat{e}$ maintains the properties of e in Lemma 14. [9] gives an efficient algorithm to compute e and $\widehat{e}$. The reason that we elect to use $\widehat{e}$ over e is because $e(P, P) = 1$, an undesirable constraint for how the bilinear maps are used obtain homomorphic properties.

3 A Circular-Secure Encryption Scheme

The following scheme is due to Boneh, Halevi, Hamburg, and Ostrovsky in [4]. We present here their scheme, with the extension (as per §4 of their paper) to the k-linear assumption (as opposed to the 1-linear assumption, i.e. DDH).

3.1 Overview

The main functionality of this ElGamal-like scheme is to blind a ciphertext by providing a long list of group elements (as opposed to only one in ElGamal), but hiding the ciphertext behind an unknown selection of those elements (in expectation, half of them). The particular selection of group elements is equivalent to the secret key. This scheme is not only circular-secure but also KDM-secure with respect to affine functions on the group used to encrypt; the underlying hardness assumption is DDH. We instead show security under k-linearity.

3.2 Description

- **Initialization.** Given security parameter λ, choose prime p such that $\frac{1}{p}$ is negligible in λ. Pick a group $\mathbf{G}$ of order p with generator g.
  Set $\ell = \lceil (k+2)\log_2 p \rceil$, $g_i \xleftarrow{R} \mathbf{G}$ for $i \in [\ell]$, and $\mathbf{s} \xleftarrow{R} \{0,1\}^\ell$. Define $h = \prod_{i=1}^{\ell} g_i^{-s_i}$. Let $pk = (g_1, \ldots, g_\ell, h)$ and $sk = (g^{s_1}, \ldots, g^{s_\ell})$.
  Publish $(p, g, \mathbf{G}, pk)$.
- **Encryption.** For (unpublished) $r \xleftarrow{R} \mathbf{Z}_p$, encrypt message $m \in \mathbf{G}$ as
$$(g_1^r, \ldots, g_\ell^r, h^r m).$$
- **Decryption.** Given ciphertext $C = (c_1, \ldots, c_\ell, d)$ and secret key $(e_1, \ldots, e_\ell)$, decrypt C to
$$d \prod_{i=1}^{\ell} c_i^{\log e_i}.$$

We call this scheme $\mathcal{E}$.

3.3 Security

Throughout the security proof, we shall switch to considering additive groups, for ease of notation.

3.3.1 Preliminary lemmata

Lemma 17. *For $c = (\alpha_1, \beta_1, \ldots, \alpha_k, \beta_k, \gamma, \delta) \leftarrow \mathcal{P}_{k\text{-}Lin}$, let*

$$M_c := \begin{pmatrix} & & \alpha_k & \gamma \\ & \iddots & & \vdots \\ \alpha_1 & & & \gamma \\ \beta_1 & \cdots & \beta_k & \delta \end{pmatrix}$$

with unspecified entries equal to 0. Then,

$$\operatorname{rank} M_c = k.$$

Note that M_c is always $(k+1) \times (k+1)$, so Lemma 17 says that M_c is not full-rank. Observe also that in the case $k = 1$, M_c appears as $\begin{pmatrix} \alpha_1 & \gamma \\ \beta_1 & \delta \end{pmatrix}$ which is precisely the (transpose of the) first block of Φ_1 in the proof of Lemma 1 in [4].

Proof. We rewrite $\alpha_i = g_i = a_i g$ and $\beta_i = r_i g_i = r_i a_i g$ for $1 \leqslant i \leqslant k$ and $\gamma = h = ag$ and $\delta = \sum_{i=1}^{k} r_i h = \sum_{i=1}^{k} r_i a g$ for $g, g_i \in \mathbf{G}$ and $a, a_i, r_i \in \mathbf{Z}_p$. Write $b_i = a_i^{-1} \in \mathbf{Z}_p$. We note then that $\sum_{i=1}^{k} a b_i \beta_i = \delta$, so if $\mathbf{c}_i$ is the ith column of M_c, from the left, then $\sum_{i=1}^{k} a b_i \mathbf{c}_i = \mathbf{c}_{k+1}$, bounding the rank above by k; since the values drawn from $\mathcal{P}_{k\text{-Lin}}$ are nonzero, we also bound the rank from below by k, by considering the α_i antidiagonal. ■

Lemma 18 (matrix k-linearity)**.** *For integers $k \leqslant r_1 < r_2 \leqslant a, b$ and $\mathcal{A} : \mathbf{G}^{a \times b} \to \{0,1\}$ a polynomial-time algorithm, where $\mathbf{G}$ is an additive group of order p, if*

$$P(\mathcal{A}, n) := \mathbf{P}\left[\mathcal{A}(\Phi) = 1 \mid \Phi \xleftarrow{R} \operatorname{rank}_n \mathbf{G}^{a \times b}\right],$$

then there is a k-linear adversary $\mathcal{B}$ that runs in approximately the same time as $\mathcal{A}$ with

$$|P(\mathcal{A}, r_2) - P(\mathcal{A}, r_1)| \leqslant (r_2 - r_1) k\text{-}LinAdv[\mathcal{B}, \mathbf{G}].$$

Proof. We proceed by hybrids on the distributions $\operatorname{rank}_n(\mathbf{G}^{a \times b})$ for $n \in [r_1, r_2] \cap \mathbf{N}$. $\mathcal{B}$ is given a k-linear challenge $(\alpha_1, \beta_1, \ldots, \alpha_{k+1}, \beta_{k+1})$. It picks $n \xleftarrow{R} [r_1 + 1, r_2] \cap \mathbf{N}$ and constructs the $a \times b$ matrix

$$\Phi' := \left(\begin{array}{c|c|c} M_c & & \\ \hline & \gamma I_{n-(k+1)} & \\ \hline & & Z \end{array}\right)$$

with $Z := 0^{(a-n)\times(b-n)}$ and the unspecified entries equal to 0. $\mathcal{B}$ selects $L \xleftarrow{R} GL_a(\mathbf{Z}_p), R \xleftarrow{R} GL_b(\mathbf{Z}_p)$ and defines $\Phi := L\Phi' R$. If the k-linear challenge was drawn from $\mathcal{P}_{k\text{-Lin}}$ then the first k rows (M_c) have rank k by Lemma 17 and so $\operatorname{rank} \Phi = n - 1$ and Φ is uniform in $\operatorname{rank}_{n-1} \mathbf{G}^{a \times b}$ (since L and R were drawn randomly); otherwise, the challenge was drawn from $\mathcal{R}_{k\text{-Lin}}$ and so $\operatorname{rank} \Phi = n$ and Φ is uniform in $\operatorname{rank}_n \mathbf{G}^{a \times b}$ (since L and R were drawn randomly).

Applying this reasoning $r_2 - r_1$ times, i.e. stepping from r_1 to r_2, each time obtaining from $\mathcal{B}$'s advantage in the k-linearity challenge a term $k\text{-LinAdv}[\mathcal{B}, \mathbf{G}]$, we obtain the desired result. ■

Definition 19 (k-universal hash family)**.** A set $\mathcal{H}$ of hashes from X to Y is a **k-universal hash family** if for all pairwise-distinct $(x_1, \ldots, x_k) \in X^k$ and all not-necessarily-pairwise-distinct $(y_1, \ldots, y_k) \in Y^k$,

$$\mathbf{P}[H(x_i) = y_i \forall 1 \leqslant i \leqslant k] = \frac{1}{|Y|^k}$$

over all $H \leftarrow \mathcal{H}$.

Definition 20 (ρ-uniformity)**.** A distribution $\mathcal{D}$ on $\mathcal{X}$ is **ρ-uniform** if

$$\sum_{x \in \mathcal{X}} \left| \mathcal{D}(x) - \frac{1}{|\mathcal{X}|} \right| \leqslant \rho.$$

Lemma 21 (simplified leftover hash lemma)**.** *$\mathcal{H}$ is a 2-universal hash family from finite sets X to Y. Then the distribution $(H, H(x))$ for $H \xleftarrow{R} \mathcal{H}, x \xleftarrow{R} X$ is $\sqrt{\frac{|Y|}{4|X|}}$-uniform on $\mathcal{H} \times Y$.*

Proof. This follows immediately from the leftover hash lemma, first proved as Lemma 4.5.1 in [7]. ■

Corollary 22. *Let $\boldsymbol{r} \xleftarrow{R} \mathbf{Z}_q^\ell$ and $\mathbf{s} \xleftarrow{R} \{0,1\}^\ell$. Then $(\boldsymbol{r}^\top \mid -\boldsymbol{r} \cdot \mathbf{s})^\top$ is $\frac{1}{q^{\frac{1}{2}(k+1)}}$-uniform in $\mathbf{Z}_q^{\ell+1}$, and hence $\frac{1}{q}$-uniform.*

Proof. For the family of hashes parameterized using $\mathbf{r} \in \mathbf{Z}_q^\ell$ given by $H_{\mathbf{r}} : \mathbf{s} \mapsto -\mathbf{r} \cdot \mathbf{s}$, this is 2-universal since the parameter space has sufficient dimension to make any output of $H_{\mathbf{r}}$ equally likely. Thus, there is a bijection between pairs $(H_{\mathbf{r}}, H_{\mathbf{r}}(\mathbf{s}))$ and vectors $(\mathbf{r}^\top | H_{\mathbf{r}}(\mathbf{s}))^\top$, so by Lemma 21, we conclude the desired bound. ■

3.3.2 The scheme $\mathcal{E}_1$

We first describe a scheme on an additive group which otherwise operates similarly to $\mathcal{E}$; accordingly, it is named $\mathcal{E}_1$.

- **Initialization.** Given security parameter λ, choose prime p such that $\frac{1}{p}$ is negligible in λ. Pick a group $\mathbf{G}$ of order p with generator g.
  Set $\ell = \lceil (k+2) \log_2 p \rceil$, $\mathbf{s} \xleftarrow{R} \{0,1\}^\ell$, $\Psi \xleftarrow{R} \operatorname{rank}_\ell \mathbf{G}^{(\ell+1)\times \ell}$, and $\Phi = (\Psi \mid -\Psi \cdot s)$. Let $pk = \Phi$ and $sk = g\mathbf{s}$.
  Publish $(p, g, \mathbf{G}, pk)$.
- **Encryption.** For (unpublished) $\mathbf{r} \xleftarrow{R} \mathbf{Z}_p^{1\times(\ell+1)}$, encrypt message $\mu \in \mathbf{G}$ as the $(\ell+1)$-vector
$$\mathbf{r} \cdot \Phi + (0^{1\times\ell} | \mu).$$
- **Decryption.** Given ciphertext $C^\top$ and secret key $\mathbf{s}$, decrypt $C^\top$ to
$$C \cdot (\mathbf{s}^\top | 1)^\top.$$

We observe that decryption holds because $(\mathbf{s}^\top|1)^\top \perp pk$, and that the only difference between $\mathcal{E}$ (when formulated over an additive group as opposed to a multiplicative group) and $\mathcal{E}_1$ is that a random combination of Φ's rows is used rather than a random multiple of the vector pk as in $\mathcal{E}$.

3.3.3 Properties of $\mathcal{E}_1$

The scheme $\mathcal{E}_1$ satisfies each of the following properties:

(a) (**secret-key homomorphism**) For invertible affine function $f : \mathbf{Z}_p^\ell \to \mathbf{Z}_p^\ell$ sending $\mathbf{x} \mapsto A\mathbf{x} + \mathbf{b}$, let $M_f = \begin{pmatrix} A & \mathbf{b} \\ 0^{1\times\ell} & 1 \end{pmatrix}$. Observe that $M_f(\mathbf{x}^\top|1)^\top = (f(\mathbf{x})^\top|1)^\top$. For $\mathbf{x} \in \{0,1\}^\ell$ a secret key for Φ, if $f(s) \in \{0,1\}^\ell$, then $\Phi \cdot M_f^{-1}$ is a public key for $f(\mathbf{s})$, and $\mathrm{Enc}_1(\Phi, \mu) \cdot M_f^{-1}$ encrypts μ with $pk = \Phi \cdot M_f^{-1}$. For instance, if $f_{\mathbf{a}} : \mathbf{s} \mapsto \mathbf{s} \oplus \mathbf{a}$ for the extension $x \oplus 0$ as the identity and $x \oplus 1 = 1 - x$, then we can easily compute corresponding public keys and ciphertexts for modifications via of a secret key by this extended XOR.

(b) (**public-key blinding**) For public key Φ and $R \xleftarrow{R} GL_{\ell+1}(\mathbf{Z}_p)$, $\Phi \cdot R$ is a uniformly random public key for Φ's secret key, and the distribution of ciphertexts under Φ is the same as that under $\Phi \cdot R$.

(c) (**self-referential encryption**) For secret key $sk = \mathbf{s}$ corresponding to public key Φ, $(g\mathbf{e}_i|0)$ is an encryption of s_i under Φ.

(d) (**plaintext homomorphism**) For affine function $f : \mathbf{G}^n \to \mathbf{G}$ sending $\mathbf{x} \mapsto \mathbf{a} + \beta$ and public key Φ, if $M \in \mathbf{G}^{n\times(\ell+1)}$ has ith row $\mathrm{Enc}(\Phi, \mu_i)$, then $\mathbf{a}M + (0^{1\times\ell}|b)$ is an encryption of $f(\mathbf{m})$

(e) (**total blinding**) For public key Φ and $\mathbf{r} \xleftarrow{R} \mathbf{Z}_p^{1\times(\ell+1)}$, $\mathbf{r}\Phi + \mathbf{c}$ is uniformly random in $\mathbf{G}^{1\times(\ell+1)}$.

3.3.4 KDM-security of $\mathcal{E}_1$

Proposition 23. *Any $\mathcal{C}_{n\ell}$-KDM-adversary $\mathcal{A}$ against $\mathcal{E}_1$ has a k-linearity adversary $\mathcal{B}$ running in about the same time for which*

$$KDM^{(n)}_{\mathcal{C}_{n\ell}}Adv[\mathcal{A}, \mathcal{E}_1] \leqslant (2\ell - 1)k\text{-}LinAdv[\mathcal{B}, \mathbf{G}] + \frac{1}{q}.$$

Proof. We shall demonstrate the result through a series of games. Let

$$w_i := \mathbf{P}[\mathcal{A} \text{ wins game } i].$$

Game 0 is the $\mathcal{C}_{n\ell}$-KDM-security game from §2.2.2, and so $\left|w_0 - \frac{1}{2}\right| = \mathrm{KDM}^{(n)}_{\mathcal{C}_{n\ell}}\mathrm{Adv}[\mathcal{A}, \mathcal{E}_1]$.

Game 1 is a modification of the $\mathcal{C}_{n\ell}$-KDM-security game, in the following way. The challenger $\mathcal{D}_1$ generates $(pk, sk) = (\Phi, \mathbf{s})$ but never uses $\mathbf{s}$ for the rest of the game, essentially "forgetting" it. Instead, picking $\mathbf{a}_i \xleftarrow{R} \{0,1\}^\ell$ for $1 \leqslant i \leqslant n$, $\mathcal{D}_1$ sends these to the adversary, which is consistent with the actual secret keys $sk_i = (\mathbf{s} \oplus \mathbf{a}_i)g$—but does not ever actually use $\mathbf{s}$. Then, $\mathcal{D}_1$ uses (a) and (b) to generate uniformly random public keys pk_i for secret keys sk_i using $\Phi, \mathbf{a}_i$.

We denote by $\mathbf{v}$ the vector $(sk_1^\top|\cdots|sk_n^\top)^\top$. For affine function $f \in \mathcal{C}_{n\ell}$, to compute $\mathrm{Enc}(pk_i, f(\mathbf{v}))$, we do the following: for each $1 \leqslant j \leqslant n$, $\mathcal{D}_1$ uses (c) to generate $\mathrm{Enc}(pk_j, \mu)$ for each $\mu \in sk_j$ and (a) to transform it into an encryption $\mathrm{Enc}(pk_i, \mu)$. Combining these into a matrix of encryptions of each element of $\mathbf{v}$ under pk_i. Using (d), $\mathcal{D}_1$ generates an encryption $\mathbf{c} \leftarrow \mathrm{Enc}(pk_i, f(\mathbf{v}))$; it then sends $\mathcal{A}$, for $\mathbf{r} \xleftarrow{R} \mathbf{Z}_p^{1\times(\ell+1)}$, $\mathbf{r}pk_i + f(\mathbf{v})$. Since the distributions are identical to those in the previous game, we conclude that $w_1 = w_0$.

Game 2 is a modification of Game 1, in the following way. $\Psi \xleftarrow{R} \operatorname{rank}_1 \mathbf{G}^{(\ell+1)\times\ell}$ and $\Phi := (\Psi \mid -\Psi\mathbf{s})$, so $\operatorname{rank}\Psi = 1$ rather than ℓ. By Lemma 18, there is a k-linearity adversary $\mathcal{B}$ running in about the same time as $\mathcal{A}$ with

$$|w_2 - w_1| \leqslant (\ell-1)k\text{-LinAdv}[\mathcal{B}, \mathbf{G}].$$

Ψ can be computed by choosing random nonzero $\psi \xleftarrow{R} \mathbf{G}^{\ell+1}, \mathbf{r} \xleftarrow{R} \mathbf{Z}_p^\ell$ and setting $\Psi := \psi\times\mathbf{r}$. By Corollary 22, $\Phi = \psi\times(\mathbf{r}^\top \mid -\mathbf{r}\cdot\mathbf{s})^\top$ is $\frac{1}{q}$-uniform in $\operatorname{rank}_1 \mathbf{G}^{(\ell+1)\times(\ell+1)}$.

Game 3 replaces Φ in Game 2 with $\Phi \xleftarrow{R} \operatorname{rank}_1 \mathbf{G}^{(\ell+1)\times(\ell+1)}$ so by $\frac{1}{q}$-uniformity, $|w_3 - w_2| \leqslant \frac{1}{q}$.

Game 4 replaces Φ in Game 3 with $\Phi \xleftarrow{R} \operatorname{rank}_{\ell+1} \mathbf{G}^{(\ell+1)\times(\ell+1)}$. By (e), the resulting ciphertexts are uniformly random, independently of $\mathcal{D}_1$'s choice of $b \leftarrow \{0,1\}$, so $w_4 = \frac{1}{2}$. By Lemma 18, there is a k-linearity adversary $\mathcal{B}$ running in about the same time as $\mathcal{A}$ with $|w_4 - w_3| \leqslant \ell \cdot k\text{-LinAdv}[\mathcal{B}, \mathbf{G}]$.

Combining the outcomes of each game, we find precisely the desired bound. ■

3.3.5 KDM-security of $\mathcal{E}$

Proposition 24. *Any $\mathcal{C}_{n\ell}$-KDM-adversary $\mathcal{A}$ against $\mathcal{E}$ has a k-linearity adversary $\mathcal{B}_1$ running in about the same time and a $\mathcal{C}_{n\ell}$-KDM adversary $\mathcal{B}_2$ also running in about the same time for which*

$$KDM^{(n)}_{\mathcal{C}_{n\ell}}Adv[\mathcal{A}, \mathcal{E}] \leqslant (\ell-1)k\text{-}LinAdv[\mathcal{B}_1, \mathbf{G}] + KDM^{(n)}_{\mathcal{C}_{n\ell}}Adv[\mathcal{B}_2, \mathcal{E}_1].$$

Proof. We shall demonstrate the result through a series of games. Let $w_i := \mathbf{P}[\mathcal{A} \text{ wins game } i]$.

Game 0 is the $\mathcal{C}_{n\ell}$-KDM-security game from §2.2.2, and so $\left|w_0 - \frac{1}{2}\right| = \text{KDM}^{(n)}_{\mathcal{C}_{n\ell}}\text{Adv}[\mathcal{A}, \mathcal{E}]$.

Game 1 is a modification of the $\mathcal{C}_{n\ell}$-KDM-security game, in the following way. The challenger $\mathcal{D}$ generates $\Psi_0 \xleftarrow{R} \operatorname{rank}_1 \mathbf{G}^{(\ell+1)\times\ell}$ and secret keys $\mathbf{s}_i \xleftarrow{R} \{0,1\}^\ell$ for $1 \leqslant i \leqslant n$. Corresponding public keys are generated from $L_i \xleftarrow{R} GL_{\ell+1}(\mathbf{Z}_p)$ and $R_i \xleftarrow{R} GL_\ell(\mathbf{Z}_p)$, using $\Psi_i := L_i\Psi_0 R_i$ and $pk_i := (\Psi_i \mid -\Psi_i\mathbf{s}_i)$. Then, for each i, $\mathcal{D}$ selects a $\mathcal{E}$ public key as any nonzero row of pk_i. (It is a valid public key for $\mathcal{E}$ by its randomness and orthogonality to $\mathbf{s}_i$.) This row ρ_i is send to $\mathcal{A}$. However, when responding to $\mathcal{A}$'s queries, $\mathcal{D}$ just uses Φ_i and $\mathcal{E}_1$, outputting for $R \xleftarrow{R} \mathbf{Z}_q^{n\times(\ell+1)}$ the ciphertext $R\Phi_i + (0|m)$ for message m. Because $\operatorname{rank}\Phi_i = 1$, each row is a scalar multiple of the others, so the distributions of public keys, secret keys and ciphertexts between Games 0 and 1 are identical. Therefore $w_1 = w_0$.

Game 2 is a modification of Game 1 with $\Psi_0 \xleftarrow{R} \operatorname{rank}_\ell \mathbf{G}^{(\ell+1)\times\ell}$, yielding valid $\mathcal{E}_1$ public key Φ. By Lemma 18, there is a k-linearity adversary $\mathcal{B}_1$ running in about the same time as $\mathcal{A}$ for which $|w_2 - w_1| \leqslant (\ell-1)k\text{-LinAdv}[\mathcal{B}_1, \mathbf{G}]$; because $\mathcal{A}$ attacks $\mathcal{E}_1$ but only being able to see one row of the public keys (an attack that we name $\mathcal{B}_2$), $\left|w_2 - \frac{1}{2}\right| = \text{KDM}^{(n)}_{\mathcal{C}_{n\ell}}\text{Adv}[\mathcal{B}_2, \mathcal{E}_1]$.

Combining the outcomes of each game, we find precisely the desired bound. ■

Combining Proposition 24 with Proposition 23 proves the security of $\mathcal{E}$.

4 A Semantically-Secure Scheme to Evaluate 2-DNF Formulas

The following scheme is due to Boneh, Goh, and Nissim in [3].

4.1 Overview

The main functionality of this ElGamal-like scheme is to blind a ciphertext with a uniformly random element in a subgroup of a semiprime-order cyclic group, in contrast to ElGamal, which blinds with a uniformly random element in a subgroup of a prime-order group. The difficulty of discerning the "component" of the ciphertext lying in this subgroup—the subgroup decision problem—underlies the scheme's security.

4.2 Description

- **Initialization.** Given security parameter λ, generate λ-bit primes q_1, q_2 and compute $n := q_1 q_2$. Generate groups $\mathbf{G}, \mathbf{G}'$ of order n with k-linear map $\widehat{e}$ sending k-tuples of elements of $\mathbf{G}$ to $\mathbf{G}'$. Pick $g, u \xleftarrow{R} \mathbf{G}$ and set $h := u^{q_2}$. The message space is $\mathcal{M} = [T]$ for some T chosen to make decryption practical. The tuple $(n, \mathbf{G}, \mathbf{G}', \widehat{e}, g, h)$ is public knowledge and constitutes the public key. The secret key is q_1.
- **Encryption.** For (unpublished) $r \xleftarrow{R} \mathbf{Z}_p$, encrypt message $m \in \mathcal{M}$ as
$$g^m h^r.$$
- **Decryption.** Given ciphertext C, compute $C^{q_1} = g^{mq_1} u^{q_1 q_2 r} = (g^{q_1})^m$. (Pre)compute g^{q_1}, and using Pollard's lambda algorithm, compute m as $\log_{g^{q_1}} C^{q_1}$.

We call this scheme $\widetilde{\mathcal{E}}$.

In [3], this is accomplished for $k = 2$ by finding the least m such that $3mn - 1$ is prime (this exists, e.g. by Dirichlet's theorem); name it p, letting $\mathbf{G}$ be the subgroup of $E(\mathbf{F}_p)$ of order n, where $E : y^2 = x^3 + 1$, and letting $\mathbf{G}'$ be the subgroup of $\mathbf{F}_{p^2}^{\times}$ of order n, where bilinear map $\widehat{e}$ is the modified Weil pairing. This setup $(n, \mathbf{G}, \mathbf{G}', \widehat{e}, g, h)$ satisfies the subgroup decision problem.

4.3 k-DNF formula evaluation

In terms of logic, a **k-disjunctive normal form formula** (k-DNF formula) takes the form
$$\bigvee_{i=1}^{n} \bigwedge_{j=1}^{n_i} X_{ij}$$
for boolean variables X_{ij} where $n_i \in [k]$ for each i. $(a \wedge \neg b) \vee c \vee (\neg d \wedge b)$ is an example of a 2-DNF formula on boolean variables. Viewing $\wedge$ as addition and $\vee$ as multiplication, on (modular) integers, a k-DNF formula takes the form
$$\sum_{i=1}^{n} \prod_{j=1}^{n_i} X_{ij}$$

again with $n_i \in [k]$. $ab + c + db$ is an example of a 2-DNF formula on (modular) integer variables.

To ensure that the above scheme can handle k-DNF formulas, we demonstrate that it can handle arbitrary additions and exactly one multiplication, so that the sum of multiplied terms can be made to match any k-DNF formula.

Addition (pre-multiplication) is multiplication of ciphertexts with rerandomization using $r \xleftarrow{R} \mathbf{Z}_n$:

$$\text{Enc}(pk, m_1)+\text{Enc}(pk, m_2) = (g^{m_1}h^{r_1})(g^{m_2}h^{r_2})h^r = g^{m_1+m_2}h^{r_1+r_2+r} = \text{Enc}(pk, m_1+m_2).$$

The sum of uniformly randomly chosen values in $\mathbf{Z}_n$ is itself uniformly random, so this encryption of $m_1 + m_2$ is also random.

Multiplication is mapping of ciphertexts under $\widehat{e}$ with rerandomization using $r \xleftarrow{R} \mathbf{Z}_n$:

$$\begin{aligned}\text{Enc}(pk, m_1) \cdot \cdots \cdot \text{Enc}(pk, m_k) &= \widehat{e}(g^{m_1}h^{r_1}, \ldots, g^{m_k}h^{r_k})h_1^r \\ &= \exp_{g_1}\left(\prod_{i=1}^{k} m_i\right)\exp_{h_1}\left(r + \sum_{d=1}^{k}\sum_{I\subset_d[k]}(aq_2)^{d-1}\prod_{i\in I} r_i \prod_{j\notin I} m_j\right)\end{aligned}$$

where $h = g^{aq_2}$ for some $a \in \mathbf{Z}_n$, $g_1 := \widehat{e}(g, g)$ and $h_1 := \widehat{e}(g, \ldots, g, h)$. In the case of $k = 2$ and $\widehat{e}$ as in §2.4, this uses the properties of $\widehat{e}$ in Lemma 14 and Lemma 16. The second line of the above equations then becomes $g_1^{m_1m_2}h_1^{m_1r_2+r_2m_1+aq_2r_1r_2+r}$. The randomness of r_i and r leads to h_1 having uniformly random exponent (whether for $k = 2$ or $k > 2$).

Addition (post-multiplication) is still multiplication of ciphertexts with rerandomization using $r \xleftarrow{R} \mathbf{Z}_n$:

$$\text{Enc}(pk, m_1)+\text{Enc}(pk, m_2) = (g_1^{m_1}h_1^{r_1})(g_1^{m_2}h_1^{r_2})h_1^r = g_1^{m_1+m_2}h_1^{r_1+r_2+r} = \text{Enc}(pk, m_1+m_2).$$

The sum of uniformly randomly chosen values in $\mathbf{Z}_n$ is itself uniformly random, so this encryption of $m_1 + m_2$ is also random.

4.4 Security

We shall show that if the initialization step satisfies the subgroup addition assumption, then the scheme is semantically secure. Suppose towards contradiction that $\mathcal{B}$ is an adversary that running in polynomial in λ time that breaks $\widetilde{\mathcal{E}}$'s semantic security, i.e. $\mathcal{B}$ wins the semantic security game with nonnegligible advantage. We shall use $\mathcal{B}$ to construct $\mathcal{A}$, running in about the same time, that breaks the supposed subgroup hardness of $\widetilde{\mathcal{E}}$'s initialization procedure.

$\mathcal{A}$ is given $(n, \mathbf{G}, \mathbf{G}', \widehat{e}, x)$ with the task to determine whether or not $x^{q_1} = 1$ (outputting 1 if so, 0 otherwise). Treating x as h, $\mathcal{A}$ gives $\mathcal{B}$ the tuple $(n, \mathbf{G}, \mathbf{G}', \widehat{e}, g, x)$, which returns $m_0, m_1 \in \mathcal{M}$. $\mathcal{A}$ draws $b \xleftarrow{R} \{0, 1\}$ and returns $g^{m_b}x^r$ to $\mathcal{B}$, which guesses $b' \in \{0, 1\}$. If $\mathcal{B}$ is correct, $\mathcal{A}$ outputs 1 in the subgroup decision game, and otherwise $\mathcal{A}$ outputs 0; this is because if $\mathcal{B}$ is correct then x is uniform in the subgroup of order q_1 in $\mathbf{G}$, and so $\mathcal{A}$ obtains $\mathcal{B}$'s same nonnegligible advantage. This suffices to prove the security of this scheme.

$\mathcal{G}$ yielding k-linear map $\widehat{e}$ is a priori no less vulnerable to a subgroup decisions adversary since n's factorization remains obscured. Therefore, a large part of the

difficulty is to find such maps in the first place. [5] discusses some of the difficulties in this task for the case $n > 2$.

5 A proposed KDM-secure scheme to evaluate k-DNF formulas

5.1 Overview

The general idea of this scheme is to exploit the encryption idea of [4]—which is convenient due to its strong security—and to exploit the homomorphism and decryption ideas of [3]. This is done by transforming the $h^r m$ term of [4] into h^{r+m}. This admits convenient addition in the obvious way and multiplication via k-linear maps.

5.2 Description

- **Initialization.** Given security parameter λ, choose prime p such that $\frac{1}{p}$ is negligible in λ. Pick a group $\mathbf{G}$ of order p and another group $\mathbf{G}'$ such that there exists a k-linear map $L : \mathbf{G}^k \to \mathbf{G}'$. Set $\ell = \lceil (k+2)\log p \rceil$, $sk = \mathbf{s} \xleftarrow{R} \{0,1\}^\ell$. $pk = (g_1, \ldots, g_\ell, h)$ for g_i random elements of $\mathbf{G}$, and $h := \prod_{i=1}^{\ell} g_i^{s_i} = \prod_{i:s_i=1} g_i$. The tuple $(p, \ell, L, \mathbf{G}, \mathbf{G}', pk)$ is public knowledge.
- **Encryption.** For (unpublished) $r \xleftarrow{R} \mathbf{Z}_p$, encrypt message $m \in \mathcal{M}$ as the $(\ell+1)$-vector
$$(g_1^r, \ldots, g_\ell^r, h^{r+m}).$$
- **Decryption.** Given ciphertext C, assume it is a matrix with dimensions
$$\underbrace{(\ell+1) \times \cdots \times (\ell+1)}_{k \text{ factors}}$$
and if instead C instead given in the form $(g_1^r, \ldots, g_\ell^r, h^{r+m})$ then consider it as
$$C \cdot \underbrace{\mathrm{Enc}(pk, 1) \cdot \cdots \cdot \mathrm{Enc}(pk, 1)}_{k-1 \text{ factors}}.$$
Let the ith of of those multiplicands be written as $(a_{i,1}, \cdots, a_{i,\ell+1})$, so c's entries are $b_{\mathbf{x}} = L(a_{1,x_1}, \ldots, a_{k,x_k})$ where $\mathbf{x} \in [\ell+1]^k$. Let

$S_d = \{b_{\mathbf{x}} \mid \mathbf{x} \in [\ell+1]^k,\ x_i = \ell+1 \text{ for exactly } d \text{ values of } i, \text{ if } x_i \neq \ell+1 \text{ then } s_{x_i} = 1\}$.

Then, compute
$$\prod_{d=0}^{k} \prod_{b \in S_d} b^{(-1)^{k+d}}. \tag{25}$$
Following [3], having chosen $\mathcal{M}$ such that a search is practical e.g. via Pollard's lambda method, we recover m.

As mentioned earlier, in [3], this is accomplished for $k = 2$ by finding the least m such that $3mn - 1$ is prime (this exists, e.g. by Dirichlet's theorem); name it p, letting $\mathbf{G}$ be the subgroup of $E(\mathbf{F}_p)$ of order n, where $E : y^2 = x^3 + 1$, and letting $\mathbf{G}'$ be the subgroup of $\mathbf{F}_{p^2}^{\times}$ of order n, where bilinear map $\widehat{e}$ is the modified Weil pairing.

5.3 Justification of Decryption

We begin by proving a lemma that will render the justification of decryption immediate.

Lemma 26. *The following algebraic identity holds:*

$$\prod_{i=1}^{k} m_i = \sum_{d=0}^{k} \sum_{I \subset_d [k]} (-1)^{k+d} \prod_{i \in I} (r_i + m_i) \prod_{j \notin I} r_j. \tag{27}$$

Proof. Fix $I_0 \subsetneq [k]$. Let $M_{I_0} := \prod\limits_{i \in I_0} m_i \prod\limits_{j \notin I_0} r_j$. We note that in the expansion of $p_I := \prod\limits_{i \in I} (r_i + m_i) \prod\limits_{j \notin I} r_j$, $\operatorname{coef}_{p_I} M_{I_0} = \begin{cases} 1 & I \supseteq I_0 \\ 0 & \text{else} \end{cases}$. We wish to use this to show that $\sum\limits_{I \subset [k]} (-1)^{k+\#I} \operatorname{coef}_{p_I} M_{I_0} = 0$. As it is clear that $\operatorname{coef}_{RHS} M_{[k]} = 1$, this will complete the proof. We observe that $\sum\limits_{I \subset [k]} (-1)^{k+\#I} \operatorname{coef}_{p_I} M_{I_0} = \sum\limits_{I_0 \subseteq I \subsetneq [k]} (-1)^{k+\#I}$. Suppose $\#I = \#I_0 + n$. The number of such sets I equals $\binom{k-\#I_0}{n}$, and so we actually have $(-1)^{k+\#I_0} \sum\limits_{n=0}^{k-\#I_0} (-1)^n \binom{k-\#I_0}{n} = 0$ by the combinatorial identity that the alternating sum along a row of Pascal's triangle vanishes. ∎

We now observe that the right-hand side of the equation in (27) is precisely the exponent of each term $L(g_{a_1}, \ldots, g_{a_k})$ for which $s_{a_i} = 1$ for $1 \leqslant i \leqslant k$ in (25).

5.4 Homomorphism

Addition (pre-multiplication) is componentwise multiplication with rerandomization using $r \xleftarrow{R} \mathbf{Z}_n$:

$$\operatorname{Enc}(pk, m_1) + \operatorname{Enc}(pk, m_2) = (g_1^{r_1+r_2+r}, \cdots, g_\ell^{r_1+r_2+r}, h^{r_1+r_2+r+m_1+m_2}) = \operatorname{Enc}(pk, m_1 + m_2).$$

The sum of uniformly randomly chosen values in $\mathbf{Z}_p$ is itself uniformly random, so this encryption of $m_1 + m_2$ is also random.

Multiplication is a mapping of all combinations under the bilinear map $L : \mathbf{G}^k \to$

$\mathbf{G}'$ to a k-fold tensor, e.g. for $k = 2$:

$$\begin{aligned}\mathrm{Enc}(pk, m_1) \cdot \mathrm{Enc}(pk, m_2) &= \begin{pmatrix} L(g_1^{r_1}, g_1^{r_2}) & \cdots & L(g_1^{r_1}, g_\ell^{r_2}) & L(g_1^{r_1}, h^{r_2+m_2}) \\ \vdots & \ddots & & \vdots \\ L(g_\ell^{r_1}, g_1^{r_2}) & & L(g_\ell^{r_1}, g_\ell^{r_2}) & L(g_\ell^{r_1}, h^{r_2+m_2}) \\ L(h^{r_1+m_1}, g_1^{r_2}) & \cdots & L(h^{r_1+m_1}, g_\ell^{r_2}) & L(h^{r_1+m_1}, h^{r_2+m_2}) \end{pmatrix} \\ &= \begin{pmatrix} L(g_1, g_1)^{r_1 r_2} & \cdots & L(g_1, g_\ell)^{r_1 r_2} & L(g_1, h)^{r_1(r_2+m_2)} \\ \vdots & \ddots & & \vdots \\ L(g_\ell, g_1)^{r_1 r_2} & & L(g_\ell, g_\ell)^{r_1 r_2} & L(g_\ell, h)^{r_1(r_2+m_2)} \\ L(h, g_1)^{(r_1+m_1) r_2} & \cdots & L(h, g_\ell)^{(r_1+m_1) r_2} & L(h, h)^{(r_1+m_1)(r_2+m_2)} \end{pmatrix}\end{aligned}$$

Addition (post-multiplication) is entrywise multiplication in the $(\ell+1) \times (\ell+1)$ matrix of elements of $\mathbf{G}'$ corresponding to encryption using the bilinear map. This is valid because each entry z of the entrywise product is of the form xy, and so

$$\begin{aligned}\prod_{d=0}^{k} \prod_{z \in S_d} z^{(-1)^{k+d}} &= \prod_{d=0}^{k} \prod_{\mathbf{i} \in I} z_{\mathbf{i}}^{(-1)^{k+d}} \\ &= \prod_{d=0}^{k} \prod_{\mathbf{i} \in I_d} (x_{\mathbf{i}} y_{\mathbf{i}})^{(-1)^{k+d}} \\ &= \left(\prod_{d=0}^{k} \prod_{\mathbf{i} \in I_d} x_{\mathbf{i}}^{(-1)^{k+d}} \right) \left(\prod_{d=0}^{k} \prod_{\mathbf{i} \in I_d} y_{\mathbf{i}}^{(-1)^{k+d}} \right) \\ &= \left(\prod_{d=0}^{k} \prod_{x \in S'_d} x^{(-1)^{k+d}} \right) \left(\prod_{d=0}^{k} \prod_{y \in S''_d} y^{(-1)^{k+d}} \right)\end{aligned}$$

where the S_d correspond to those sets in the product, S'_d, S''_d to those sets in each multiplicand, and I_d to the indices of matrix elements giving rise to S_d.

After performing any of these homomorphic operations, we re-randomize by multiplying each entry of the form a^b (where a is known, b may not be known) by a^r for randomly chosen r. e.g., for multiplication in the case $k = 2$, the final matrix would be

$$\begin{pmatrix} L(g_1, g_1)^{r_1 r_2 + r} & \cdots & L(g_1, g_\ell)^{r_1 r_2 + r} & L(g_1, h)^{r_1(r_2+m_2)+r} \\ \vdots & \ddots & & \vdots \\ L(g_\ell, g_1)^{r_1 r_2 + r} & & L(g_\ell, g_\ell)^{r_1 r_2 + r} & L(g_\ell, h)^{r_1(r_2+m_2)+r} \\ L(h, g_1)^{(r_1+m_1) r_2 + r} & \cdots & L(h, g_\ell)^{(r_1+m_1) r_2 + r} & L(h, h)^{(r_1+m_1)(r_2+m_2)+r} \end{pmatrix}.$$

5.5 Security

We shall show that the above scheme $\mathcal{E}'$ is n-way KDM-secure with respect to $\mathcal{C}_{n\ell}$. Suppose towards contradiction that adversary $\mathcal{A}'$ has nonnegligible advantage $\mathrm{KDMAdv}^{(n)}_{\mathcal{C}_{n\ell}}[\mathcal{A}', \mathcal{E}'](\lambda)$ and achieves this by sending the challenger the pairs (i_k, g'_k) for $1 \leqslant k \leqslant N$, $1 \leqslant i_k \leqslant n$ and $g_k \in \mathcal{C}_{n\ell}$ (not to be confused with the selector functions f_k). We now construct an adversary $\mathcal{A}$ for $\mathcal{E}$ with the same advantage. $\mathcal{A}$ constructs the functions $g_k : \mathbf{G}^{n\ell} \to \mathbf{G}$ given by $g_k(x_1, \ldots, x_{n\ell}) =$

$\exp_h(g'_k(\log x_1, \ldots, \log x_{n\ell}))$. Then, $\mathcal{A}$'s queries are of the form (i_k, g_k), and receives the outputs c_k from the $\mathcal{E}$-challenger. We note that these outputs correspond precisely to original encryptions with the functions g'_k if the challenger chose $b = 0$ and 0 (corresponding to the element $1 \in \mathbf{G}$) otherwise. $\mathcal{A}$ then gives the challenger's outputs to $\mathcal{A}'$, which outputs b', and so $\mathcal{A}$ also outputs b' having the same (nonnegligible) advantage as $\mathcal{A}'$, a violation of the hardness assumptions. Thus the desired security follows.

6 Future work

One substantial drawback to this scheme is that ciphertext sizes (after application of the k-linear map) are exponential in k, namely $\lceil (k+2)\log p \rceil^k$. [4] provides one means to shorten ciphertext sizes in its §4, but only by a factor of $O(\log\log p)^k$—making ciphertext lengths still $\omega(k^k)$. One possible future route would be to find a way to keep ciphertexts at size subexponential in k, perhaps even linear or constant.

Another more ambitious objective is to use this idea to generate fully homomorphic encryption. This would likely only be possible after reducing ciphertext sizes to a constant in k; then, depending on the (circuit) complexity of the decryption method, it might be possible to pass that decryption circuit through the encryption circuit, effectively bootstrapping the scheme into a fully homomorphic encryption.

Of course, a main hurdle to all of this is in finding k-linear maps to begin with, for $k > 2$. A discussion of this can be found in [5], where a high degree of difficulty is posited for this task. Thus this is a question that must first be resolved before the scheme here may be brought to fruition.

References

[1] Dan Boneh, Xavier Boyen, Hovav Shacham. "Short Group Signatures." In: *Proceedings of Crypto* 3152 (2004), pp.41–55.

[2] Dan Boneh, Matthew Franklin. "Identity-Based Encryption from the Weil Pairing." In: *SIAM Journal of Computing* 32.3 (2003), pp.586–615.

[3] Dan Boneh, Eu-Jin Goh, Kobbi Nissim. "Evaluating 2-DNF Formulas on Ciphertexts." In: *Proceedings of Theory of Cryptography* (2005), pp.325–341.

[4] Dan Boneh, Shai Halevi, Mike Hamburg, Rafail Ostrovksy. "Circular-Secure Encryption from Decision Diffie-Hellman." In: *Proceedings of Cryptography* (2008), pp.108–125.

[5] Dan Boneh and Alice Silverberg. "Applications of Multilinear Forms to Cryptography." In: *Contemporary Mathematics* 324 (2003), pp.71–90.

[6] Craig Gentry. "A Fully Homomorphic Encryption Scheme." PhD thesis, Stanford University (2009).

[7] Johan Hastad, Russell Impagliazzo, Leonid A. Levin, Michael Luby. "A pseudorandom generator from any one-way function." In: *SIAM Journal on Computing*, 28.4 (1999), pp.1364–1396.

[8] Dennis Hofheinz and Eike Kiltz. "Secure Hybrid Encryption from Weakened Key Encapsulation." In: *Proceedings of CRYPTO* (2007), pp.553–571.

[9] Victor S. Miller. "The Weil Pairing, and Its Efficient Computation." In: *Journal of Cryptography* 17 (2004), pp.235–261.

[10] John M. Pollard. "Monte Carlo Methods for Index Computation (mod p)." In: *Mathematics of Computation* 32.143 (1978), pp.918–924.

[11] Hovav Shacham. "A Cramer-Shoup Encryption Scheme from the Linear Assumption and from Progressively Weaker Linear Variants." In: *Cryptology ePrint Archive*, 2007.

[12] Joseph H. Silverman. *The Arithmetic of Elliptic Curves* 2ed. Springer Graduate Texts in Mathematics, 2009.