The Multivariate Hidden Number Problem

Steven D. Galbraith and Barak Shani

Department of Mathematics, University of Auckland, New Zealand

Abstract

This work extends the line of research on the hidden number problem. Motivated by studying bit security in finite fields, we define the multivariate hidden number problem. Here, the secret and the multiplier are vectors, and partial information about their dot product is given. Using tools from discrete Fourier analysis introduced by Akavia, Goldwasser and Safra, we show that if one can find the significant Fourier coefficients of some function, then one can solve the multivariate hidden number problem for that function. This allows us to generalise the work of Akavia on the hidden number problem with (non-adaptive) chosen multipliers to all finite fields.

We give two further applications of our results, both of which generalise previous works to all (finite) extension fields. The first considers the general (random samples) hidden number problem in $\mathbb{F}_{p^m}$ and assumes an advice is given to the algorithm. The second considers a model that allows changing representations, where we show hardness of individual bits for elliptic curve and pairing based functions for elliptic curves over extension fields, as well as hardness of any bit of any component of the Diffie-Hellman secret in $\mathbb{F}_{p^m}$ $(m > 1)$.

Keywords: hidden number problem, bit security, hardcore bits

1. INTRODUCTION

The computational Diffie-Hellman assumption (CDH) states that for appropriate groups G, given values $g, g^a, g^b \in G$, the Diffie-Hellman secret g^{ab} is hard to compute. However, this assumption does not rule out the possibility that some bits of g^{ab} are predictable. This leads to interesting theoretical questions about the security of bits arising from computational problems. A useful language to express these ideas is the *hidden number problem*. Informally, the hidden number problem in a (multiplicative) group G with a (non-constant) function f defined over G is the problem of recovering a hidden element $s \in G$ given pairs $(t_i, f(st_i))$.

This problem was introduced by Boneh and Venkatesan [7] in order to study bit security (specifically blocks of most-significant bits) of the Diffie-Hellman secret. They were the first

to prove hardness of bits for Diffie-Hellman key exchange. Today, this problem is studied in its own right and is of theoretical interest, and also leads to practical results, outside the scope of the Diffie-Hellman key exchange (see, for example, [10, 14]). It is most desirable to prove security of the smallest possible blocks of bits (i.e., blocks of size 1).

Interested in the hidden number problem in (finite) extension fields, we study the following variant of the hidden number problem, which we call the *multivariate hidden number problem*. Here, the problem takes place over a ring R, on which a function f is defined, and the secret $\mathbf{s} = (s_1, \ldots, s_m)$ is an m-tuple in R^m. Informally again, the problem is recovering the secret $\mathbf{s}$ given pairs $(\mathbf{t}_i, f(\mathbf{s} \cdot \mathbf{t}_i))$, where $\mathbf{s} \cdot \mathbf{t}_i$ is the dot product of $\mathbf{s}$ and $\mathbf{t}_i$. That is, $f(\mathbf{s} \cdot \mathbf{x}) = f(s_1x_1 + \cdots + s_mx_m)$ for $\mathbf{x} = (x_1, \ldots, x_m) \in R^m$.

This problem arises naturally from the following observation. Assume an oracle $\mathcal{O}$ gives partial information, e.g. one bit, of one (fixed) component of sx, for a secret s and a multiplier x in $\mathbb{F}_{p^m}$. One would like to learn s. First, the component can be expressed as a dot product $\widetilde{\mathbf{s}} \cdot \mathbf{x}$ for $\mathbf{x} \in (\mathbb{F}_p)^m$, a vector that represents x, and some $\widetilde{\mathbf{s}} \in (\mathbb{F}_p)^m$. If one can learn $\widetilde{\mathbf{s}}$ given $\mathcal{O}(x) = bit(\widetilde{\mathbf{s}} \cdot \mathbf{x})$, then the learner can solve the hidden number problem by computing s from $\widetilde{\mathbf{s}}$.

Previous Work

The hidden number problem has been extensively studied, and different variants have been proposed throughout the years, as well as numerous extensions (for a comprehensive overview of the different extensions, see Shparlinski's survey [18]). Boneh and Venkatesan [7] considered $G = \mathbb{Z}_p^*$ for prime p and showed that the $\sqrt{\log p} + \log\log p$ most-significant bits of the Diffie-Hellman secret g^{ab} are as hard to compute as the whole secret. Their approach uses lattice basis reduction. There is a considerable subsequent literature, including the case of extension fields, but lattice methods are unable to obtain hardness results for single bits.

Significant progress resulted from the introduction of tools from Fourier analysis (learning theory) by Akavia, Goldwasser and Safra [3] (for a complete description, see Akavia's thesis [1]). They showed that if one can find the *heavy* Fourier coefficients of a function, then one can solve the hidden number problem for that function. In addition, they built on the fundamental work of Goldreich and Levin [13] and Kushilevitz and Mansour [15] and provided an algorithm to find heavy Fourier coefficients of a function, under the *membership queries model*. This new approach allows to consider hardness of single bits, even for noisy oracles that only have a non-negligible advantage over the bias of the function in question. Since these tools work under specific query-access models, they can only be used to solve the hidden number problem when the solver has the suitable access to the function.

This new approach, involving Fourier analysis, laid the groundwork for subsequent interesting results in the study of bit security. Akavia [2] gave a solution to the hidden number problem with chosen multipliers in the multiplicative group of prime fields $\mathbb{F}_p$ for a family of functions, called *concentrated* functions, where multipliers are chosen *non-adaptively*[1]. Akavia also showed that the most-significant-bit function is concentrated. Morillo and Ràfols [16] proved that, for any integer $1 \leq k \leq \log_2(N)$, the k-th bit function on $\mathbb{Z}_N$ is concentrated (they specifically considered N a prime or an RSA modulus). This can be combined with Akavia's result on concentrated functions.

By combining the above with the work of Boneh and Shparlinski [6], Duc and Jetchev [9] showed the hardness of any single bit of elliptic curve and pairing based functions for elliptic curves over prime fields, in a model that allows the solver to change the representation of the group. In a similar model, Fazio, Gennaro, Perera and Skeith [12] gave the first single bit hardness result for Diffie-Hellman secrets in an extension field – excluding hardness of the constant-term component bits – where they considered the field $\mathbb{F}_{p^2} = \mathbb{F}_p[x]/(x^2 + Ax + B)$ (with p a prime) using a polynomial basis representation. A very recent result by Wang, Zhan and Zhang [20] generalised this work to extension fields $\mathbb{F}_{p^m}$, where m is polynomial in $\log p$. As in [12], only polynomial basis representations are considered in [20].

Our Contribution

Our contribution is first and foremost of a mathematical nature. We show that if one can find heavy Fourier coefficients of a function f, then one can solve the multivariate hidden number problem for f. This is done by proving an algebraic relation between the Fourier transforms of f and $f_{\mathbf{s}}$, where $f_{\mathbf{s}}(\mathbf{x}) := f(\mathbf{s} \cdot \mathbf{x})$ as above. Using the algorithm from [1, 3], we give a solution to a chosen-multiplier version of the multivariate hidden number problem for concentrated functions f over $\mathbb{F}_p$, where multipliers are chosen non-adaptively.

This allows us to generalise the solution to the hidden number problem with chosen multipliers to all finite fields $\mathbb{F}_{p^m}$ for concentrated functions, which include the k-th bit function of each component, for every $1 \leq k \leq \log_2(p)$.

We also give several application of our main results. We show how the results can be used in different models, one of which is the "representation changing" model. By constructing isomorphisms between representations of $\mathbb{F}_{p^m}$ that forms a dot product (as in the multivariate hidden number problem) in a specific component, we show that changing field representations gives the required multipliers needed to solve the multivariate hidden number problem, for concentrated functions over $\mathbb{F}_p$. Specifically, we prove hardness of any single bit of any

[1] As noted in [7], if we let "the queries be correlated" the problem already had a known solution for a block of one bit "even when the oracle is noisy".

component for Diffie-Hellman secrets in $\mathbb{F}_{p^m}$. We do not restrict only to polynomial representations. This result holds for general vector space representations and also normal basis representations of $\mathbb{F}_{p^m}$. We also give bit security results for elliptic curve and pairing based functions for elliptic curves over $\mathbb{F}_{p^m}$.

We stress that as with previous work our results are not sufficient to prove (single) bit security of the classic Diffie-Hellman key exchange. This is due to the fact that the chosen multipliers needed for these approaches cannot be obtained when attacking the Diffie-Hellman protocol. However, one can obtain bit security results for Diffie-Hellman and related schemes by considering algorithms with advice, as was done by Akavia [2], for example.

Paper Organisation

The paper is organised as follows. Section 2 gives definitions and some facts needed for our later results. Sections 3 and 4 are our main theoretical contributions. In section 3 we introduce the multivariate hidden number problem and establish our main tool, to be used in Section 4, where we give our main results: solutions to the multivariate hidden number problem over $\mathbb{F}_p$ and the hidden number problem in $\mathbb{F}_{p^m}$. Section 5 focuses on other applications. We discuss two models in which our results can be applied, by giving the appropriate background and summarizing recent results. We then show how one can use our results to prove bit security in these models, and how it relates to previous work.

2. PRELIMINARIES

2.1 Fourier Analysis on Finite Groups

Let $(R, +, \cdot)$ be a finite ring and denote by $G := (R, +)$ the corresponding additive abelian group. We are interested in the set of functions $\{f : R \to \mathbb{C}\}$. This set of functions is a vector space (over the complex field), whose dimension is $|R|$, since, for instance, the Kronecker delta functions $\{\delta_i\}_{i \in R}$ $\left(\delta_i(j) = 1 \text{ if } j = i, \text{ otherwise } \delta_i(j) = 0\right)$ form a basis for this vector space; every function $f : R \to \mathbb{C}$ can be written as $f(x) = \sum_{i \in R} f(i)\delta_i(x)$. Let $\overline{z}$ denote the complex conjugate of a complex number z. We define an inner product in this vector space by $\langle f, g \rangle := \mathbb{E}_{x \in R}\left[f(x) \cdot \overline{g(x)}\right] = \frac{1}{|R|}\sum_{x \in R} f(x) \cdot \overline{g(x)}$. The l_2 norm of a function f is $\|f\|_2 := \sqrt{\langle f, f \rangle}$.

A character of G is a group homomorphism taking values in the non-zero complex numbers, namely $\chi : G \to \mathbb{C}^*$ such that $\chi(x + y) = \chi(x)\chi(y)$. Since $\chi(x)^{|G|} = \chi(|G| \cdot x) = \chi(0_G) = 1$, we get that the characters take values in the complex $|G|$-th roots of unity. Moreover, there are exactly $|G|$ of them, so we associate each character χ to a group element $a \in G$,

yielding χ_a. That is, denote by $\widehat{G}$ the set (group) of characters of G, and consider the map $\varphi : G \to \widehat{G}$, given by $\varphi(a) := \chi_a$. The map φ can be shown to be an isomorphism.

An alternative basis for $\{f : R \to \mathbb{C}\}$ is the Fourier basis consisting of all the characters χ. Standard facts in Fourier analysis on finite groups are: for the trivial character $\chi_0 \in \widehat{G}$ it holds that $\sum_{x\in G} \chi_0(x) = |G|$, and $\sum_{x\in G} \chi(x) = 0$ if $\chi_0 \neq \chi \in \widehat{G}$; in addition, these characters are orthogonal and have l_2 norm of 1, hence the Fourier basis is an orthonormal basis. Therefore, we can represent each function $f : R \to \mathbb{C}$ as a linear combination of the characters χ_a. This linear combination is given by $f(x) = \sum_{a\in G} \widehat{f}(a)\chi_a(x)$, where each coefficient is the Fourier transform $\widehat{f}(a) := \langle f, \chi_a\rangle$. Let $\overline{\chi}_a$ be the conjugate to the character χ_a. That is, $\overline{\chi}_a(x) = \overline{\chi_a(x)}$.

For $G = \mathbb{Z}_N$ we define the characters χ_a by $\chi_a(x) := \mathrm{e}^{\frac{2\pi i}{N}ax}$. For $G = \mathbb{Z}_{N_1} \times ... \times \mathbb{Z}_{N_m}$, let $\mathbf{a} = (a_1, \ldots, a_m)$ and $\mathbf{x} = (x_1, \ldots, x_m)$; the character $\chi_{\mathbf{a}}(\mathbf{x})$ is given by $\chi_{\mathbf{a}}(\mathbf{x}) := \chi_{a_1}(x_1) \cdot ... \cdot \chi_{a_m}(x_m) = \mathrm{e}^{\frac{2\pi i}{N_1}a_1x_1} \cdot ... \cdot \mathrm{e}^{\frac{2\pi i}{N_m}a_mx_m}$.

Let $f : \mathbb{Z}_p \to \mathbb{C}$ and define the function $f_s : \mathbb{Z}_p \to \mathbb{C}$ by $f_s(x) := f(sx)$, for $s \in \mathbb{Z}_p^*$. The well-known scaling property of the Fourier transform is the following relation between the Fourier transforms (with respect to the additive group $G = (\mathbb{Z}_p, +)$) of f and f_s: $\widehat{f}_s(z) = \widehat{f}(zs^{-1})$. This is a basic property of the Fourier transform, which follows from the fact that $\chi_z(sx) = \chi_{zs}(x)$. This relation inspires our approach in Lemma 13, and so we see fit to show its proof.

Lemma 1. *Let $s \in \mathbb{Z}_p^*$, let $f : \mathbb{Z}_p \to \mathbb{C}$ and define $f_s : \mathbb{Z}_p \to \mathbb{C}$ by $f_s(x) := f(sx)$ for every $x \in \mathbb{Z}_p$. The Fourier transform of f_s satisfies $\widehat{f}_s(z) = \widehat{f}(zs^{-1})$ for every $z \in \mathbb{Z}_p$.*

Proof. By definition of the Fourier transform we get that

$$\widehat{f}_s(z) = \frac{1}{p}\sum_{x\in\mathbb{Z}_p} f_s(x)\overline{\chi}_z(x) = \frac{1}{p}\sum_{x\in\mathbb{Z}_p} f(sx)\overline{\chi}_z(x) \ .$$

Since $x' := sx$ is a permutation of $\mathbb{Z}_p$, we change the order of summation and sum over x'. Therefore,

$$\begin{aligned}
\widehat{f}_s(z) &= \frac{1}{p}\sum_{x'} f(x')\overline{\chi}_z(s^{-1}x') \\
&= \frac{1}{p}\sum_{x'} f(x')\mathrm{e}^{-\frac{2\pi i}{p}z(s^{-1}x')} = \frac{1}{p}\sum_{x'} f(x)\mathrm{e}^{-\frac{2\pi i}{p}(zs^{-1})x'} \\
&= \frac{1}{p}\sum_{x'} f(x')\overline{\chi}_{zs^{-1}}(x') = \widehat{f}(zs^{-1}) \ .
\end{aligned}$$

■

We now recall some definitions from [3, 9, 16]. The same definitions can be made for functions over rings R where G is their additive group.

Definition 2 (Restriction)**.** Given a function $f : G \to \mathbb{C}$ and a set of characters Γ, the *restriction of f to* Γ is the function $f|_\Gamma : G \to \mathbb{C}$ defined by $f|_\Gamma := \sum_{\chi_a \in \Gamma} \widehat{f}(a)\chi_a$.

Definition 3 (Concentration)**.** A function $f : G \to \mathbb{C}$ is *Fourier concentrated* if for every $\epsilon > 0$ there exist a set Γ of $poly\Big(\log\Big(\frac{|G|}{\epsilon}\Big)\Big)$ characters, such that $\|f - f|_\Gamma\|_2^2 \leq \epsilon$.

Definition 4 (Heavy coefficient)**.** For a function $f : G \to \mathbb{C}$ and a threshold $\tau > 0$, we say that a coefficient $\widehat{f}(a)$ (corresponding to the character χ_a) is *τ-heavy* if $|\widehat{f}(a)|^2 > \tau$.

Theorem 5 (Akavia [1])**.** *There is a probabilistic algorithm that given a finite group G, a threshold $\tau > 0$ and oracle query access to a function $f : G \to \mathbb{C}$, finds all the τ-heavy Fourier coefficients. The algorithm runs in polynomial time in* $\log\big(|G|\big)$, $\frac{1}{\tau}$ *and* $\|f\|_2$.

The models of oracle access in this paper are discussed in Remark 8 below.

2.2 Finite Field Representations

Let $\mathbb{F}$ be a finite field. A known fact is that if $\mathbb{F}$ has q elements, then q is a power of some prime p, that is, $q = p^m$ for a prime p and a positive integer m. Hence, we denote that field by $\mathbb{F}_q$. Another known fact is that given a number $q = p^m$ as above, there is a unique field with q elements, up to isomorphism. Yet, $\mathbb{F}_q$ has different (all isomorphic to each other) representations. One representation of a field $\mathbb{F}_{p^m}$ is given by $\mathbb{F}_p[x]/\,(h)$, where $\mathbb{F}_p[x]$ is the ring of polynomials with coefficients in $\mathbb{F}_p$, the polynomial h is a monic irreducible polynomial of degree m in $\mathbb{F}_p[x]$, and (h) is the principal ideal generated by h. We emphasize that there are also other representations, like the normal basis $\{\theta, \theta^p, \theta^{p^2}, \ldots, \theta^{p^{m-1}}\}$, where θ is an element of the field such that this set is linearly independent, and $\theta^{p^m} = \theta$.

The field $\mathbb{F}_{p^m}$ is a vector space of dimension m over the field $\mathbb{F}_p$, equipped with a bilinear inner product. For an aribtrary vector space basis of $\mathbb{F}_{p^m}$ there are m^3 structure coefficients which determine the multiplication rule in $\mathbb{F}_{p^m}$. For completeness we state and prove the following standard result.

Lemma 6. *Let $\{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m\}$ be a basis of the vector space $\mathbb{F}_{p^m}$ over $\mathbb{F}_p$. For elements $u, v \in \mathbb{F}_{p^m}$, let $\boldsymbol{u}, \boldsymbol{v}$ be the coefficient vectors in $\mathbb{F}_p^m$ corresponding to this vector space basis. There exist m invertible matrices $M_1, \ldots, M_m$ such that $uv = \sum_{k=1}^m w_k \boldsymbol{b}_k$, where each coefficient is given by $w_k = \boldsymbol{u} M_k \boldsymbol{v}^T$.*

Proof. For a basis $\{\mathbf{b}_1, \ldots, \mathbf{b}_m\} \subseteq \mathbb{F}_{p^m}$, the structure coefficients determine the product of all the basis elements. That is, $\mathbf{b}_i \mathbf{b}_j = \sum_{k=1}^m c_{i,j}^k \mathbf{b}_k$, where $c_{i,j}^k$ are the structure coefficients. Then, by the bilinearity of multiplication, we get that a product of any two elements

$u = \sum_{i=1}^{m} u_i \mathbf{b}_i$ and $v = \sum_{j=1}^{m} v_j \mathbf{b}_j$ is of the form

$$uv = \sum_{i=1}^{m} u_i \sum_{j=1}^{m} v_j \left(\mathbf{b}_i \mathbf{b}_j\right) = \sum_{i=1}^{m} u_i \sum_{j=1}^{m} v_j \sum_{k=1}^{m} c_{i,j}^{k} \mathbf{b}_k = \sum_{k=1}^{m} \sum_{i=1}^{m} u_i \widetilde{v_i^k} \mathbf{b}_k \ ,$$

where $\widetilde{v_i^k}$ is a linear combination of the scalars v_i with $c_{i,j}^k$ as coefficients. In other words, by representing the multiplication of **u** and **v** as linear combination of the basis elements – $\mathbf{uv} = \sum_{k=1}^{m} w_k \mathbf{b}_k$ – every coefficient w_k in this linear combination is of the linear form $u_1 \widetilde{v_1^k} + ... + u_m \widetilde{v_m^k}$, where u_i are the coefficients of **u**. The existence of the matrices M_k follows.

Assume that M_k is not invertible, then there exists $\mathbf{u} \neq 0$ such that $\mathbf{u} M_k = 0$. Hence, for every **v**, the coefficient $w_k = 0$. Let $u \neq 0$ be the field element corresponding to **u**. We get that multiplication by u is not an injection. Therefore u is a zero divisor – a contradiction. ■

2.3 Hidden Number Problem

The hidden number problem was introduced in [7] in order to study the bit security of Diffie-Hellman key exchange. The relation between the two is explained in Remark 9 below. The problem was introduced over the multiplicative group $\mathbb{Z}_p^*$, but it can be generalised to arbitrary finite (abelian) groups. Since our applications involve single bit functions, we present the problem with a single bit function.

Definition 7 (Hidden number problem (single bit))**.** Let $(G, \cdot)$ be a group, let $s \neq 0$ be a secret element of G and let $f : G \to \{-1, 1\}$. The goal is to find the secret element s using oracle access to the function $f_s(x) := f(sx)$.

Remark 8 (Access models)**.** We use the term *oracle access* as a general term for any of the following oracle models. We follow the language from [17] in describing the oracle access models in this paper. When we write *query access* we refer to the *membership queries model*, where the learner can query the function on any input $x \in G$ and receive the sample $(x, f_s(x))$. In the *uniform distribution model*, the learner has access to a random source of samples: at each time the learning algorithm queries, a random input $x \in G$ is chosen uniformly, and the sample $(x, f_s(x))$ is returned to the algorithm.

Models of HNP. We adopt the notation from [6] and write HNP-CM for a chosen-multiplier version of the hidden number problem, which is under the membership queries model. That is, in HNP-CM the learner can query the function on any input. We emphasize that in this paper, unlike [6], any queries in this model are made *non-adaptively*. This means that the algorithm

first chooses all its queries, and after receiving the response starts its process. This is opposed to adaptive queries, where the queries may depend on the secret s and are adjusted during the process of recovering s. When a solver can choose multipliers adaptively the problem already has a solution (based on the work in [5], and later [4]).

In the original (more general) variant of the hidden number problem, which we denote by HNP, the oracle access is in the uniform distribution model. That is, the solver only gets pairs $(t_i, f_s(t_i))$, for d elements $t_1, \ldots, t_d \in G$ chosen independently and uniformly at random. This is probably the most frequently discussed variant of the hidden number problem.

Unfortunately, the algorithm in Theorem 5 cannot be used in the uniform distribution model, and therefore cannot be used to solve HNP. The upside of Theorem 5 is that it is strong enough to handle oracles that only have a non-negligible advantage over the bias of the function in question. That is, the results hold even for a noisy oracle, i.e., an oracle that does not give a correct answer all the time, but with some probability. Since this work focuses on a mathematical framework, we do not elaborate on this noise model. The interested reader can look at [1, 3, 9, 16].

Remark 9. One historical motivation for the hidden number problem is the following. Given a group G, an element g in the group and the values g^a and g^b, the shared Diffie-Hellman secret s is the value $s = DH_g(g^a, g^b) = g^{ab}$. Notice that one can choose a number k and calculate g^k, then by multiplying g^k and g^a, one gets $g^a g^k = g^{a+k}$. An active attacker in the static Diffie-Hellman protocol (where Bob always uses a fixed value g^b), who has access to some bit of the shared secret, can send the value g^{a+k} to Bob, so that Bob calculates the value $(g^{a+k})^b = g^{ab}g^{bk} = sg^{bk}$ and we notice that the attacker can calculate the value g^{bk} by $(g^b)^k$, yielding the (uniformly distributed) multiplier for the secret (in the hidden number problem). The attacker's goal (computing s) is exactly the hidden number problem.

An alternative interpretation is to consider a Diffie-Hellman oracle. Suppose we have an oracle that on input g^x and g^y outputs some bits of g^{xy}. We can query this oracle on g^b and g^{a+k} for several k's, and if we can solve the hidden number problem, we can find the secret $s = g^{ab}$.

Terminology. Adopting the language from [2], we say that an algorithm (l, δ, t)-solves the (multivariate) hidden number problem if the number of queries to the oracle is at most l, the algorithm outputs the hidden number s with probability at least δ, and the running time is at most t. We say that an algorithm *solves* the (multivariate) hidden number problem if $\frac{1}{\delta}$, l and t are polynomials in $\log\big(|G|\big)$.

We now recall the main result of Akavia [2] and sketch its proof. A full proof can be found in [2] (with a different terminology of the hidden number problem; for more details see our discussion in Section 5.1). We divide the result into two parts. Theorem 10 shows that an algorithm that learns heavy Fourier coefficients of functions over $\mathbb{F}_p$, leads to a solution to the hidden number problem in $\mathbb{Z}_p^*$. Corollary 11 shows how to solve (with non-adaptive queries) HNP-CM for concentrated functions in $\mathbb{Z}_p$. The ability to choose multipliers in HNP-CM is what allows one to have the oracle query access needed in applying the algorithm from Theorem 5, which allows to solve the hidden number problem.

Theorem 10 ([2])**.** *Let $\mathcal{A}$ be an algorithm that learns the τ-heavy Fourier coefficients of functions defined over $\mathbb{F}_p$. For any concentrated[2] function $f : \mathbb{F}_p \to \{-1, 1\}$, there exists an algorithm that solves the hidden number problem in $\mathbb{Z}_p^*$.*

Proof sketch. Let $f_s : \mathbb{F}_p \to \{-1, 1\}$ be the function from the hidden number problem, i.e., $f_s(x) := f(sx)$. By the scaling property (Lemma 1) we know that the Fourier coefficients of f_s are simply the Fourier coefficients of f permuted by s^{-1}. One might imagine that it is easy to compute the lists of Fourier coefficents of both f and f_s and then match them up to deduce the permuting element s^{-1}. However, this is not an efficient task when p is large (in both aspects: computing and comparing). This is where the idea of using concentrated functions is crucial. Instead of computing all the Fourier coefficients we just locate the τ-heavy ones for suitable τ, using the learning algorithm $\mathcal{A}$ on both f and f_s. These lists are short (by Parseval) and so matching up the values to find the permutation factor s^{-1} is efficient. ∎

Corollary 11 ([2])**.** *For any concentrated[3] function $f : \mathbb{F}_p \to \{-1, 1\}$, there exists an algorithm that solves HNP-CM in $\mathbb{Z}_p^*$, where the queries are made non-adaptively.*

Akavia proved that the most-significant-bit function $MSB : \mathbb{Z}_p \to \{-1, 1\}$ is concentrated, and hence proved that HNP-CM in $\mathbb{Z}_p^*$ with the MSB function can be solved. Later on, Morillo and Ràfols [16] proved that, for any integer $1 \leq k \leq \log_2(p)$, the k-th bit function on $\mathbb{Z}_p$ is concentrated. Therefore, HNP-CM in $\mathbb{Z}_p^*$ can also be solved with these functions.

3. Multivariate Hidden Number Problem

In this section we define our variant of the hidden number problem, which we call the multivariate hidden number problem, and then introduce the tool that helps us solve this problem.

[2] In [2], a different definition of concentration is taken. We use the definition from [3]. Both papers use the same method to obtain the proof of Theorem 10.

[3] See previous footnote.

Definition 12 (Multivariate hidden number problem (single bit))**.** Let R be a ring, let $\mathbf{s} = (s_1, \ldots, s_m) \neq (0, \ldots, 0)$ be a secret in R^m, and let $f : R \to \{-1, 1\}$. The goal is to find the secret $\mathbf{s}$ using oracle access to the function $f_{\mathbf{s}}(\mathbf{x}) := f(\mathbf{s} \cdot \mathbf{x}) = f(s_1x_1 + \cdots + s_mx_m)$.

For $m = 1$ the multivariate hidden number problem is simply the hidden number problem. As noted in [18], a polynomial version of the hidden number problem (poly-HNP) can be considered. This polynomial version can be seen as a special case of the multivariate hidden number problem. As above, we write MV-HNP-CM for a chosen-multiplier version of the multivariate hidden number problem, and MV-HNP for uniformly random multipliers.

The following lemma gives a relation between the Fourier transforms of $f_{\mathbf{s}}$ and f, analogous to the relation in Lemma 1 (scaling property). This lemma may be of independent interest.

Lemma 13. *Let $f : \mathbb{Z}_p \to \mathbb{C}$, let $\boldsymbol{s} = (s_1, \ldots, s_m) \in \mathbb{Z}_p^m$ be such that not all $s_i = 0$, and define $f_{\boldsymbol{s}} : \mathbb{Z}_p^m \to \mathbb{C}$ by $f_{\boldsymbol{s}}(\boldsymbol{x}) := f(\boldsymbol{s} \cdot \boldsymbol{x})$. For any $s_k \neq 0$, the Fourier transform of $f_{\boldsymbol{s}}$ satisfies*

$$\widehat{f_{\boldsymbol{s}}}(z_1, \ldots, z_m) = \begin{cases} \widehat{f}(z_k s_k^{-1}) & \text{if } \; z_j - z_k s_k^{-1} s_j = 0, \;\; \forall 1 \leq j \neq k \leq m; \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

Proof. Recall that a character in $\mathbb{Z}_p$ is defined by $\chi_a(x) = \mathrm{e}^{\frac{2\pi i}{p}ax}$ and that for an element $\mathbf{a} = (a_1, \ldots, a_m) \in \mathbb{Z}_p^m$ the character $\chi_{\mathbf{a}}(\mathbf{x})$ is given by $\chi_{\mathbf{a}}(\mathbf{x}) = \prod_{i=1}^m \chi_{a_i}(x_i)$. Therefore, for $1 \leq k \leq m$, we have

$$\begin{aligned} \chi_{(a_1,\ldots,a_m)}(x_1, \ldots, x_m) &= \prod_{i=1}^{m} \chi_{a_i}(x_i) = \prod_{i \neq k} \chi_{a_i}(x_i)\, \chi_{a_k}(x_k) \\ &= \chi_{(a_1,\ldots,a_{k-1},a_{k+1},a_m)}(x_1, \ldots, x_{k-1}, x_{k+1}, x_m)\chi_{a_k}(x_k) \;\; . \end{aligned}$$

Assume without loss of generality that $s_m \neq 0$. Then, $\widehat{f_{\mathbf{s}}}(z_1, \ldots, z_m) =$

$$\begin{aligned} & \frac{1}{p^m} \sum_{(x_1,\ldots,x_m) \in \mathbb{Z}_p^m} f_{\mathbf{s}}(x_1, \ldots, x_m)\, \overline{\chi}_{(z_1,\ldots,z_m)}(x_1, \ldots, x_m) \\ = \;& \frac{1}{p^m} \sum_{x_1,\ldots,x_m \in \mathbb{Z}_p} f(s_1x_1 + \cdots + s_mx_m)\, \overline{\chi}_{(z_1,\ldots,z_m)}(x_1, \ldots, x_m) \\ = \;& \frac{1}{p^m} \sum_{x_1,\ldots,x_{m-1}} \sum_{x_m} f(s_1x_1 + \cdots + s_mx_m)\, \overline{\chi}_{(z_1,\ldots,z_{m-1})}(x_1, \ldots, x_{m-1})\, \overline{\chi}_{z_m}(x_m) \\ = \;& \frac{1}{p^m} \sum_{x_1,\ldots,x_{m-1}} \overline{\chi}_{(z_1,\ldots,z_{m-1})}(x_1, \ldots, x_{m-1}) \sum_{x_m} f(s_1x_1 + \cdots + s_mx_m)\, \overline{\chi}_{z_m}(x_m) \;\; . \end{aligned}$$

Since $x'_m := s_m x_m$ is a permutation of $\mathbb{Z}_p$, we change the order of summation and sum over x'_m. Therefore, $\widehat{f_{\mathbf{s}}}(z_1, \ldots, z_m) =$

$$\frac{1}{p^m} \sum_{x_1,\ldots,x_{m-1}} \overline{\chi}_{(z_1,\ldots,z_{m-1})}(x_1, \ldots, x_{m-1}) \sum_{x'_m} f\left(s_1 x_1 + \cdots + s_{m-1} x_{m-1} + x'_m\right) \overline{\chi}_{z_m}(s_m^{-1} x'_m) \ .$$

Let $y := s_1 x_1 + \cdots + s_{m-1} x_{m-1} + x'_m$, so that $f(s_1 x_1 + \cdots + s_{m_1} x_{m_1} + x'_m) = f(y)$. We get that $\widehat{f_{\mathbf{s}}}(z_1, \ldots, z_m) =$

$$\begin{aligned}
& \frac{1}{p^{m-1}} \sum_{x_1,\ldots,x_{m-1}} \overline{\chi}_{(z_1,\ldots,z_{m-1})}(x_1, \ldots, x_{m-1}) \cdot \\
& \frac{1}{p} \sum_{y} f(y) \overline{\chi}_{z_m}\left(s_m^{-1}(y - s_1 x_1 - \cdots - s_{m-1} x_{m-1})\right) \\
= \ & \frac{1}{p^{m-1}} \sum_{x_1,\ldots,x_{m-1}} \overline{\chi}_{(z_1,\ldots,z_{m-1})}(x_1, \ldots, x_{m-1}) \cdot \\
& \frac{1}{p} \sum_{y} f(y) \overline{\chi}_{z_m s_m^{-1}}(y) \, \overline{\chi}_{\left(-z_m s_m^{-1} s_1, \ldots, -z_m s_m^{-1} s_{m-1}\right)}(x_1, \ldots, x_{m-1}) \\
= \ & \frac{1}{p^{m-1}} \sum_{x_1,\ldots,x_{m-1}} \overline{\chi}_{\left(z_1 - z_m s_m^{-1} s_1, \ldots, z_{m-1} - z_m s_m^{-1} s_{m-1}\right)}(x_1, \ldots, x_{m-1}) \, \widehat{f}\left(z_m s_m^{-1}\right) \\
= \ & \widehat{f}\left(z_m s_m^{-1}\right) \frac{1}{p^{m-1}} \sum_{x_1,\ldots,x_{m-1}} \overline{\chi}_{\left(z_1 - z_m s_m^{-1} s_1, \ldots, z_{m-1} - z_m s_m^{-1} s_{m-1}\right)}(x_1, \ldots, x_{m-1}) \ .
\end{aligned}$$

The last sum equals 0 unless the character $\chi_{(z_1 - z_m s_m^{-1} s_1, \ldots, z_{m-1} - z_m s_m^{-1} s_{m-1})}$ is the trivial character in $\mathbb{Z}_p^{m-1}$, in which case it equals p^{m-1}.[4] Therefore we get that $\widehat{f_{\mathbf{s}}}(z_1, ..., z_m) = \widehat{f}(z_m s_m^{-1})$ when $z_j - z_m s_m^{-1} s_j = 0$ for all $1 \leq j \leq m-1$ and otherwise $\widehat{f_{\mathbf{s}}}(z_1, ..., z_m) = 0$, as stated in (1). ∎

The interesting property of $f_{\mathbf{s}}(\mathbf{x})$ is that its Fourier coefficients are equal to zero outside the line $(x_1, \ldots, x_m) = (ts_1, \ldots, ts_m)$ for $t \in \mathbb{Z}_p$. Along this line the Fourier coefficients of $f_{\mathbf{s}}(\mathbf{x})$ are those of $f(x)$. So it is like the graph of the Fourier spectrum of $f(x)$ is drawn along a diagonal line in the space $\mathbb{Z}_p^m$.

We now give our main tool that allows to attack the multivariate hidden number problem using Fourier learning. Denote by $Heavy_\tau(f) = \{c_i \ : \ |\widehat{f}(c_i)|^2 > \tau\}$ the list that represents all τ-heavy Fourier coefficients of f.

Proposition 14. *Let $f : \mathbb{Z}_p \to \{-1, 1\}$, let $\boldsymbol{s} = (s_1, \ldots, s_m) \in \mathbb{Z}_p^m$ be such that not all $s_i = 0$, and let $f_{\boldsymbol{s}} : \mathbb{Z}_p^m \to \{-1, 1\}$ be the function $f_{\boldsymbol{s}}(\boldsymbol{x}) := f(\boldsymbol{s} \cdot \boldsymbol{x})$. Then, $Heavy_\tau(f) = \{c_1, \ldots, c_t\}$*

[4] Recall that $\sum_{x \in G} \chi(x) = 0$ if $\chi_0 \neq \chi \in \widehat{G}$, and for the trivial character $\chi_0 \in \widehat{G}$ we get $\sum_{x \in G} \chi_0(x) = |G|$.

if and only if $Heavy_\tau(f_{\boldsymbol{s}}) = \{(c_i s_1, \ldots, c_i s_m) \mid 1 \leq i \leq t\}$. *In other words, a coefficient* $\widehat{f_s}(z_1, \ldots, z_m)$ *of* $f_{\boldsymbol{s}}$ *is* τ*-heavy if and only if there exists* $1 \leq i \leq t$ *such that* $z_j = c_i s_j$ *for every* $1 \leq j \leq m$ *and* $\widehat{f}(c_i)$ *is* τ*-heavy.*

Proof. The claim follows from Lemma 13. Let $1 \leq k \leq m$ such that $s_k \neq 0$. Assume $c \in Heavy_\tau(f)$ and consider the vector $(z_1, \ldots, z_m) = (cs_1, \ldots, cs_m)$. Specifically $z_k = cs_k$, so $c = z_k s_k^{-1}$ and therefore for every $1 \leq j \leq m$ one gets $z_j = cs_j = z_k s_k^{-1} s_j$ or $z_j - z_k s_k^{-1} s_j = 0$. From Lemma 13 we get that $\widehat{f_s}(cs_1, \ldots, cs_m) = \widehat{f_s}(z_1, \ldots, z_m) = \widehat{f}(z_k s_k^{-1}) = \widehat{f}(c)$. Therefore, we get that $(cs_1, \ldots, cs_m) \in Heavy_\tau(f_{\mathbf{s}})$. That is,

$$|\widehat{f}(c)|^2 > \tau \Longrightarrow |\widehat{f_{\mathbf{s}}}(cs_1, \ldots, cs_m)|^2 > \tau \ .$$

Conversely,

$$\begin{aligned} |\widehat{f_s}(z_1, \ldots, z_m)|^2 > \tau \ &\Longrightarrow \widehat{f_s}(z_1, \ldots, z_m) \neq 0 \\ &\Longrightarrow z_j = z_k s_k^{-1} s_j \text{ for every } 1 \leq j \leq m \\ &\Longrightarrow z_j = cs_j \text{ for } c = z_k s_k^{-1} \in \mathbb{Z}_p \\ &\Longrightarrow \widehat{f}(c) = \widehat{f}(z_k s_k^{-1}) = \widehat{f_s}(z_1, \ldots, z_m) \\ &\Longrightarrow |\widehat{f}(c)|^2 > \tau \ . \end{aligned}$$

That is, the coefficient $\widehat{f_{\mathbf{s}}}(z_1, \ldots, z_m)$ is τ-heavy if and only if there exists $1 \leq i \leq t$ such that $z_j = c_i s_j$ for every $1 \leq j \leq m$ and $\widehat{f}(c_i)$ is τ-heavy. ∎

Corollary 15. *Let* f *be a function defined over* $\mathbb{Z}_p$*, let* $\boldsymbol{s} = (s_1, \ldots, s_m) \in \mathbb{Z}_p^m$ *be a secret, and let* $f_{\boldsymbol{s}}$ *be a function over* $\mathbb{Z}_p^m$ *defined by* $f_{\boldsymbol{s}}(\boldsymbol{x}) := f(\boldsymbol{s} \cdot \boldsymbol{x})$*. The function* f *is concentrated if and only if the function* $f_{\boldsymbol{s}}$ *is concentrated.*

Proof. Let Γ be a set of characters of $\mathbb{Z}_p$, and define $\Gamma_s := \{\chi_{\mathbf{a}} \mid \mathbf{a} = (as_1, \ldots, as_m), \chi_a \in \Gamma\}$ to be the corresponding set of characters of $\mathbb{Z}_p^m$. The proof is evident, since $\sum_{\mathbf{a} \in \Gamma_s} |\hat{f}_{\mathbf{s}}(\mathbf{a})|^2 = \sum_{a \in \Gamma} |\hat{f}(a)|^2$. ∎

4. Main Results

In this section we show that an algorithm that learns heavy Fourier coefficients of functions over finite abelian groups, leads to solutions to the multivariate hidden number problem over $\mathbb{F}_p$ and to the hidden number problem in $\mathbb{F}_{p^m}$.

Theorem 16. *Let* $\mathcal{A}$ *be an algorithm that learns the* τ*-heavy Fourier coefficients of functions defined over finite abelian groups. For any concentrated function* $f : \mathbb{F}_p \to \{-1, 1\}$*, there exists an algorithm that solves the multivariate hidden number problem over* $\mathbb{F}_p$*.*

Proof. The proof follows from Proposition 14 and the proof of Theorem 10. Since the function f is concentrated, we can run the learning algorithm $\mathcal{A}$ (on f) in the group $\mathbb{Z}_p$. When p is very small we can just compute the list of all Fourier coefficients. When p is large we can experiment with the learning algorithm (in polynomial time) to choose a suitable threshold τ, so that one can obtain in polynomial time in $\log(p)$ a short list of τ-heavy coefficients of f.

From Corollary 15, the function $f_{\mathbf{s}}$ is concentrated, so running the learning algorithm $\mathcal{A}$ (on $f_{\mathbf{s}}$ with the same threshold τ) in the group $\mathbb{Z}_p^m$ outputs in polynomial time in $\log(p^m) = m\log(p)$ the list of τ-heavy coefficients of $f_{\mathbf{s}}$. We use the relation between the (τ-heavy) coefficients of $f_{\mathbf{s}}$ and f from Proposition 14 and follow the same process from the proof of Theorem 10 to recover the secrets $s_1, \ldots, s_m$. ∎

Since the algorithm from Theorem 5 can learn heavy Fourier coefficients for functions over arbitrary finite fields in the membership queries model, even in the presence of noise, we get the following:

Corollary 17. *For any concentrated function $f : \mathbb{F}_p \to \{-1, 1\}$, there exists an algorithm that solves MV-HNP-CM over $\mathbb{F}_p$, where the queries are made non-adaptively.*

Proof. Take $\mathcal{A}$ to be the algorithm from Theorem 5 and apply Theorem 16. ∎

We turn from the multivariate hidden number problem to the hidden number problem. Recall that the hidden number problem in the group $(R^*, \cdot)$ considers the multiplication in R, and not the dot product used in the multivariate hidden number problem. We now consider $R = \mathbb{F}_{p^m}$ as a vector space. Given a basis of $\mathbb{F}_{p^m}$ we represent an element $a \in \mathbb{F}_{p^m}$ by its components vector (related to the given basis): $\mathbf{a} = (a_1, \ldots, a_m)$. We use Lemma 6 to show that for every $1 \leq i \leq m$, the i-th component of the product as (for $a, s \in \mathbb{F}_{p^m}$) can be represented as $\mathbf{a}M_i\mathbf{s}^T$, where M_i is an invertible matrix. Therefore, for a function F over $\mathbb{F}_{p^m}$ we have $F_s(a) := F(sa) = F(\mathbf{a}M_1\mathbf{s}^T, \ldots, \mathbf{a}M_m\mathbf{s}^T)$. Note that this is a general property of $\mathbb{F}_{p^m}$ as a vector space, and therefore applies to all types of field representation. Hence, the following theorem can be applied for normal bases, polynomial bases or any other vector space basis for $\mathbb{F}_{p^m}$.

Theorem 18. *Let $\mathcal{A}$ be an algorithm that learns the τ-heavy Fourier coefficients of functions over finite abelian groups. Fix $1 \leq i \leq m$, and let $f : \mathbb{F}_p \to \{-1, 1\}$ be a concentrated function. For any function $F : \mathbb{F}_{p^m} \to \{-1, 1\}$ given by $F(\boldsymbol{x}) = F(x_1, \ldots, x_m) := f(x_i)$, there exists an algorithm that solves the hidden number problem in $\mathbb{F}_{p^m}$.*

Proof. Let $s \in \mathbb{F}_{p^m}$ be the secret element in the hidden number problem, written as $\mathbf{s} = (s_1, \ldots, s_m)$ with respect to any vector space basis of $\mathbb{F}_{p^m}$. Fix $1 \leq i \leq m$ and consider

the i-th component in $\mathbb{F}_p^m$. Lemma 6 shows that for each multiplier $a \in \mathbb{F}_{p^m}$ (written as $\mathbf{a} = (a_1, \ldots, a_m)$) in the hidden number problem we can represent the i-th component of the product as by $\mathbf{a}M_i\mathbf{s}^T = \sum_{j=1}^m a_j\widetilde{s_j}$, where $\widetilde{s_j} := \left(M_i\mathbf{s}^T\right)_j$. Therefore $F_s(x) := F(sx) = f(\widetilde{s_1}x_1 + ... + \widetilde{s_m}x_m \bmod p)$. Thus, oracle access to $F_s(x)$ is equivalent to oracle access to $f(\widetilde{s_1}x_1 + ... + \widetilde{s_m}x_m \bmod p)$. The latter is the multivariate hidden number problem over $\mathbb{F}_p$ with the concentrated function f. By Theorem 16 we can solve this problem to retrieve $\widetilde{\mathbf{s}} = (\widetilde{s_1}, \ldots, \widetilde{s_m})$. Since the matrix M_i is invertible, and since $\mathbf{a}M_i\mathbf{s}^T = \sum_{j=1}^m a_j\widetilde{s_j} = \mathbf{a} \cdot \widetilde{\mathbf{s}}^T$, we can recover the secret $\mathbf{s}$ by $\mathbf{s}^T = M_i^{-1}\widetilde{\mathbf{s}}^T$, that is, $\mathbf{s} = \widetilde{\mathbf{s}}\left(M_i^{-1}\right)^T$. ∎

Corollary 19. *Fix $1 \leq i \leq m$, and let $f : \mathbb{F}_p \to \{-1, 1\}$ be a concentrated function. For any function $F : \mathbb{F}_{p^m} \to \{-1, 1\}$ given by $F(\boldsymbol{x}) = F(x_1, \ldots, x_m) := f(x_i)$, there exists an algorithm that solves HNP-CM in $\mathbb{F}_{p^m}$, where the queries are made non-adaptively.*

Remark 20. One should notice that, having the ability to query the function at specific points, one can easily reduce the m-dimensional problem to m one-dimensional instances, then solve them one-by-one using back substitution of previous parts that were recovered. This is in fact how the algorithm from Theorem 5 works over direct product of groups.

Remark 21. We stress that our methods do not hold for the elliptic-curve-based hidden number problem. One of the reasons that these methods do not work in the elliptic curve case is that, unlike $\mathbb{F}_{p^m}^*$, the elliptic curve group law in $E(\mathbb{F}_q)$ is not of a bilinear form $\mathbf{s} \cdot \mathbf{x}$.

5. APPLICATIONS

In this section we give several applications, under different models, of our main results. These applications generalise previous bit security results to all extension fields. In Section 5.1 we generalise the work of Akavia [2] on the hidden number problem in prime fields. In Section 5.2 we generalise the works of Fazio et al. [12] on bit security of CDH in $\mathbb{F}_{p^2}$ and of Duc and Jetchev [9] on hardness of individual bits of elliptic curve and pairing based functions for elliptic curve over prime fields. We show how to reduce each problem to the form of MV-HNP-CM. The bit security results follow from the solutions given in the previous section.

5.1 Solving the Hidden Number Problem in $\mathbb{F}_{p^m}^*$ with Multipliers of the Form g^x Using Advice

The idea of using advice to solve different variants of the hidden number problem was first considered by Boneh and Venkatesan [8]. Using advice bits, independent of the secret s, they were able to solve the hidden number problem with uniformly random samples in prime fields

$\mathbb{F}_p$ for a function that outputs the $2 \log \log p$ most-significant bits. Shparlinski and Winterhof [19] modified this work to extend the result to certain subgroups of $\mathbb{F}_p$, also under the provided advice.

The terminology of Corollary 11 above is slightly different than given in [2]. There, the following variant of the hidden number problem is considered: the solver chooses values x and the multipliers for the secret s are of the form g^x. This is the original formulation of the hidden number problem in [7], which has in mind attacks on Diffie-Hellman key exchange (see Remark 9 above for more details). Clearly, this problem is harder than HNP-CM, since one has to solve certain discrete logarithms (to the base g) in order to be able to choose the right multipliers. For this reason an additional advice was considered in [2]. This short advice depends only on p and g (and not on the secret s) – it is exactly certain discrete logarithms.

Since Corollary 19 is a generalisation of Corollary 11 to extension fields, our results hold for this variant of the hidden number problem. That is, we get the following result.

Corollary 22. *Let $1 \leq i \leq m$, let $f : \mathbb{F}_p \to \{-1, 1\}$ be concentrated. For any function $F : \mathbb{F}_{p^m} \to \{-1, 1\}$ given by $F(\boldsymbol{x}) = F(x_1, \ldots, x_m) := f(x_i)$, there exists an algorithm that solves with advice the hidden number problem with multipliers of the form g^x in the group $\mathbb{F}_{p^m}^*$.*

As shown in [8] and then discussed in [2, 19] this result can be applied to show bit security of ElGamal's public key system and Okamoto's conference key sharing scheme.

5.2 Hardness of Every Single Bit of CDH by Changing Representations

Diffie-Hellman key exchange and many other cryptographic protocols can be considered for $\mathbb{F}_{p^m}$ with $m > 1$. Hence, it is of interest to consider bit security results in that context. It is also interesting to consider bit security for elliptic curve groups $E(\mathbb{F}_q)$.

The idea of changing representations to show hardness of bits of Diffie-Hellman secrets was first considered in [6] for Weierstrass equations of elliptic curves (defined over prime fields). They show the hardness of the least-significant bit of a Diffie-Hellman secret S in $E(\mathbb{F}_p)$ under a very strong model, in which the solver not only gets the value $f(S)$ (and therefore the value $f(S + P)$ for points $P \in E(\mathbb{F}_p)$, as explained in Remark 9 above),[5] but also gets the values $f(\phi(S))$, where $\phi(S)$ is the image of the point under an elliptic curve isomorphism $\phi : E(\mathbb{F}_p) \to E'(\mathbb{F}_p)$ to a different Weierstrass model, for isomorphisms that can be chosen by the solver. This idea was followed in [9] for elliptic curves defined over prime

[5] In [6] the function f is the least-significant-bit function $LSB : \mathbb{Z}_p \to \{-1, 1\}$.

fields, where hardness of single bits of elliptic-curve-based functions is considered, and in [12, 20] for extension fields in polynomial basis representation, where hardness of single bits of (polynomial-represented) Diffie-Hellman secrets is considered.

5.2.1 CDH in $\mathbb{F}_{p^m}$ with Field Isomorphisms

In [12] the field $\mathbb{F}_{p^2}$ is considered. Succinctly, when one considers the leading-coefficient component (coefficient of x), they show how one can choose multipliers by taking appropriate field isomorphisms to another polynomial basis. Let $K = k_1 x + k_0 \in \mathbb{F}_{p^2}$ be unknown (recall that $k_1, k_0 \in \mathbb{F}_p$), and suppose one is interested to learn the secret value k_1. Any isomorphism $\phi : \mathbb{F}_{p^2} \to \mathbb{F}_{p^2}$ of polynomial representations of the finite field maps K to $\phi(K) = \lambda_1 k_1 x + \lambda_0 k_1 + k_0$, for $\lambda_1, \lambda_0 \in \mathbb{F}_p$. Therefore, in a model for which one has oracle access to a single bit of the x-component after any such chosen isomorphism (therefore, chosen λ_1, λ_0), we get HNP-CM in $\mathbb{F}_p$.

The case of the constant-term component (coefficient of x^0) was left open, as well as the case of extension fields $\mathbb{F}_{p^m}$ where $m > 2$. In [20] some steps are taken to close this gap. They generalise the result of [12] to extension fields $\mathbb{F}_{p^m}$.[6] This is done by similar methods that give rise to HNP-CM in $\mathbb{F}_p$, where the secret is one of the components k_i. As in [12], the constant-term component k_0 is excluded.

For the case in which $K = g^{ab}$ is a Diffie-Hellman secret in $\mathbb{F}_{p^m}$, one can use the results involving summing functions from [11] and recover the entire secret K from the algorithm that recovers a single (fixed) component k_i.

Remark 23. Such models give some assurance that bits in the Diffie-Hellman protocol are hard. The results can be interpreted as follows: considering Diffie-Hellman key exchange over an elliptic curve (resp. a finite field), specific bits of the secret key cannot be easy to compute for all (in fact, for a non-negligble fraction of) the representations of the elliptic curve (resp. polynomial representations of the field) at once. That is, given the bit we wish to compute, there exists a representation for which this bit is hard to compute. However, this model does not prove anything about a fixed representation of the elliptic curve or finite field. It does not give any assurance of hardness of a specific bit of a specific group representation.

We show that under the representation changing model in arbitrary extension fields $\mathbb{F}_{p^m}$ one can recover directly the secret K using our solution to MV-HNP-CM. For $K = g^{ab}$, a Diffie-Hellman secret in $\mathbb{F}_{p^m}$, this shows hardness of any bit of any component, under the specified

[6] In [20] they specifically consider $\mathbb{F}_{p^m}$ where m is polynomial in $\log p$. They also show that if oracle access to a single bit of the constant-term component in $\mathbb{F}_{p^2}$ is given, then one can recover the secret value k_1.

model. This result improves the results of Fazio et al. [12] and Wang et al. [20] by showing a direct reduction from the computational Diffie-Hellman assumption, with no intermediate steps. In addition, the result holds for all extension fields $\mathbb{F}_{p^m}$. This allows us to consider the case of large m, and in particular the case of fields with small characteristic.

Moreover, we do not restrict only to polynomial bases. The result holds for general vector spaces and normal bases. Polynomial bases are more restrictive and do not allow to recover the entire secret directly, since the isomorphisms restrict the multipliers of the constant term. Note that there is no particular reason to choose polynomial bases to represent $\mathbb{F}_{p^m}$, so we recommend to use normal bases to get efficient field arithmetic and the strongest bit security result.

We now state our result for the Diffie-Hellman protocol. Note that in fact this result holds for any secret element in $\mathbb{F}_{p^m}$.

Corollary 24. *Let $s = DH_g(g^a, g^b) = g^{ab}$ be a Diffie-Hellman secret in $\mathbb{F}_{p^m}$. Given $g, g^a, g^b \in \mathbb{F}_{p^m}$, computing a single bit of s in a random vector space or normal bases representation of $\mathbb{F}_{p^m}$ is as hard as computing s. In other words, an algorithm that has a non-negligible advantage over a random guess in computing a single bit of s (in a random vector space or normal bases representation of $\mathbb{F}_{p^m}$) can be used to efficiently compute s.*

Proof. Let $\mathbf{s} \in \mathbb{F}_{p^m}$ be a secret with components $s_1, \ldots, s_m \in \mathbb{F}_p$. Assume one has oracle access to a bit of component j of elements in $\mathbb{F}_{p^m}$. Given $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_m) \in \mathbb{F}_p^m$, one needs to construct an isomorphism $\phi_\lambda^j : \mathbb{F}_{p^m} \to \mathbb{F}_{p^m}$ between representations of the finite field such that component j of $\phi_\lambda^j(\mathbf{s})$ is of the form $\lambda_1 s_1 + \cdots + \lambda_m s_m$. The result then follows.

Recall that $\mathbb{F}_{p^m}$ is a vector space of dimension m over the field $\mathbb{F}_p$, that has different types of representations. We briefly discuss the construction of a suitable isomorphism in the cases of interest.

General vector space $\mathbb{F}_p^m$. Let $B_1 = \{v_1, \ldots, v_m\}, B_2 = \{u_1, \ldots, u_m\}$ be two bases of $\mathbb{F}_p^m$. The mapping ϕ_λ^j of an element $\mathbf{s} = s_1 v_1 + \cdots + s_m v_m$ should satisfy

$$\phi_\lambda^j(\mathbf{s}) = (*)u_1 + \cdots + (\lambda_1 s_1 + \lambda_2 s_2 + \cdots + \lambda_m s_m)u_j + \cdots + (\star)u_m \ .$$

Consider this linear map as a matrix. One can easily see that the j-th row of this matrix should be $(\lambda_1\ ,\ \lambda_2\ ,\ \ldots\ ,\ \lambda_m)$. In order for the matrix to be a full rank map – therefore an isomorphism – it should be nonsingular. One can easily construct such a linear map.

Normal basis. Let $B_1 = \{\alpha, \alpha^p, \ldots, \alpha^{p^{m-1}}\}, B_2 = \{\beta, \beta^p, \ldots, \beta^{p^{m-1}}\}$ be two normal bases of $\mathbb{F}_{p^m}$. The mapping ϕ_λ^j of an element $\mathbf{s} = s_1\alpha + \cdots + s_m\alpha^{p^{m-1}}$ should satisfy

$$\phi_\lambda^j(\mathbf{s}) = (*)\beta + \cdots + (\lambda_1 s_1 + \lambda_2 s_2 + \cdots + \lambda_m s_m)\beta^{p^{j-1}} + \cdots + (\star)\beta^{p^{m-1}} \ . \qquad (2)$$

Consider the linear map satisfying $\phi^j_\lambda(\alpha) = \lambda_j\beta + \lambda_{j-1}\beta^p + \cdots + \lambda_{j+1}\beta^{p^{m-1}}$ (indices for λ_k are taken modulo m such that $1 \leq k \leq m$, i.e., $\lambda_0 = \lambda_m$ and $\lambda_{m+1} = \lambda_1$). Then

$$\begin{aligned}
\phi^j_\lambda(\mathbf{s}) &= \phi^j_\lambda(s_1\alpha + \cdots + s_m\alpha^{p^{m-1}}) = s_1\phi^j_\lambda(\alpha) + s_2\phi^j_\lambda(\alpha)^p + \cdots + s_m\phi^j_\lambda(\alpha)^{p^{m-1}} \\
&= s_1(\lambda_j\beta + \lambda_{j-1}\beta^p + \cdots + \lambda_{j+1}\beta^{p^{m-1}}) \\
&\quad + s_2(\lambda_j\beta + \lambda_{j-1}\beta^p + \cdots + \lambda_{j+1}\beta^{p^{m-1}})^p + \cdots \\
&\quad\quad + s_m(\lambda_j\beta + \lambda_{j-1}\beta^p + \cdots + \lambda_{j+1}\beta^{p^{m-1}})^{p^{m-1}} \\
&= s_1(\lambda_j\beta + \lambda_{j-1}\beta^p + \cdots + \lambda_{j+1}\beta^{p^{m-1}}) \\
&\quad + s_2(\lambda_j\beta^p + \lambda_{j-1}\beta^{p^2} + \cdots + \lambda_{j+1}\beta) + \cdots \\
&\quad\quad + s_m(\lambda_j\beta^{p^{m-1}} + \lambda_{j-1}\beta + \cdots + \lambda_{j+1}\beta^{p^{m-2}}) \ ,
\end{aligned}$$

where the last equality follows from $\beta^{p^m} = \beta$ for normal bases. After collecting the terms for each β^{p^k} (with $0 \leq k \leq m-1$) one gets (2). In order for ϕ^j_λ to be an isomorphism, one needs to check that $\phi^j_\lambda(\alpha)^{p^m} = \phi^j_\lambda(\alpha)$ and that the set $\{\phi^j_\lambda(\alpha), \phi^j_\lambda(\alpha)^p, \ldots, \phi^j_\lambda(\alpha)^{p^{m-1}}\}$ is linearly independent. This can be easily shown: the former property follows from $\beta^{p^m} = \beta$, while the latter from the linear independence of the basis B_2. ■

Remark 25 (Polynomial basis)**.** Given a polynomial $a = a_m x^{m-1} + \cdots + a_2 x + a_1$, one looks for an isomorphism ϕ^j_λ such that

$$\phi^j_\lambda(a) = (*)x^{m-1} + \cdots + (\lambda_1 a_1 + \lambda_2 a_2 + \cdots + \lambda_m a_m)x^{j-1} + \cdots + (\star)x^0 \ .$$

For the constant polynomial $1 = 0 \cdot x^{m-1} + \cdots + 0 \cdot x + 1$ one gets that the coefficient of x^{j-1} of the polynomial $\phi^j_\lambda(1)$ is λ_1, i.e., $\phi^j_\lambda(1) = \lambda_1 x^{j-1} + \ldots$. Since an isomorphism maps the identity element to the identity element, it follows that if $j \neq 1$, then λ_1 has to be 0, and if $j = 1$, then λ_1 has to be 1. Therefore, when using polynomial representations, one cannot choose multipliers for s_1 and therefore cannot recover the secret s_1 using the solution to MV-HNP-CM. One can still try to recover some, or all, of the other coefficients using the method to solve MV-HNP-CM. We leave it for future work – it is an open problem to construct isomorphisms that give rise to the required multipliers even for some coefficients.

5.2.2 CDH in $E(\mathbb{F}_{p^m})$ with Changing Weierstrass/ Field Representations

Let E be an elliptic curve over a field $\mathbb{F}_{p^m}$ and let $S = (s_x, s_y) \in E(\mathbb{F}_{p^m})$ be a secret point. We wish to learn S using oracle access to some function on changed representations of S. Such results have applications for CDH and pairing functions on elliptic curves.

The simplest approach is to assume we can get a bit of a component of s_x under changes of field representation. The result then follows from the methods of Section 5.2.1.

If we cannot change the field representation, then we can change the Weierstrass equation as was done by Boneh and Shparlinski [6]. Suppose E is given by the Weierstrass Equation $W : y^2 = x^3 + Ax + B$. For a non-zero $\lambda \in \mathbb{F}_{p^m}$ let $W_\lambda : Y^2 = X^3 + A\lambda^4 X + B\lambda^6$. The map $\phi_\lambda : W \to W_\lambda$ that takes $P = (x, y)$ on W to $P_\lambda = (\lambda^2 x, \lambda^3 y)$ on W_λ is known to be an isomorphism of groups. The image of the point $S = (s_x, s_y) \in W$ under ϕ_λ is $\phi_\lambda(S) = (\lambda^2 s_x, \lambda^3 s_y)$.

One can see that if t is a quadratic residue in $\mathbb{F}_{p^m}$, that is $t = \lambda^2$ for some $\lambda \in \mathbb{F}_{p^m}$, then by considering only the x-coordinate, the function ϕ_λ allows to choose multipliers for the secret. That is, $\phi_\lambda(S)_x = s_x t$, where $s_x, t \in \mathbb{F}_{p^m}$. Therefore, changing Weierstrass equations allows to choose multipliers for the secret, as long as t is a quadratic residue in $\mathbb{F}_{p^m}$. Due to the work in [9] for elliptic curves defined over prime fields, this is sufficient to solve HNP-CM in $\mathbb{F}_{p^m}$. A similar approach holds for the y-coordinate. Using the solution to HNP-CM in $\mathbb{F}_{p^m}$ given in Corollary 19, this forms bit security results as in [9] for elliptic curves defined over extension fields $E(\mathbb{F}_{p^m})$. Since the solution to HNP-CM in $\mathbb{F}_{p^m}$ given in Corollary 19 does not take into account the representation of the field $\mathbb{F}_{p^m}$, this result holds for any such representation.

These arguments give the following result, which we state for the Diffie-Hellman protocol, but can also be stated for elliptic-curve-based one-way functions and pairing-based one-way functions as in [9], and in fact holds for any secret element in $E(\mathbb{F}_{p^m})$.

Corollary 26. *Let $S = DH_P([a]P, [b]P) = [ab]P$ be a Diffie-Hellman secret in $E(\mathbb{F}_{p^m})$. Given $P, [a]P, [b]P \in E(\mathbb{F}_{p^m})$, computing a single bit of S for a random representation of $E(\mathbb{F}_{p^m})$ or a random representation of $\mathbb{F}_{p^m}$ is as hard as computing S. In other words, an algorithm that has a non-negligible advantage over a random guess in computing a single bit of S (for a random representation of $E(\mathbb{F}_{p^m})$ or a random representation of $\mathbb{F}_{p^m}$) can be used to efficiently compute S.*

Acknowledgements

We thank the anonymous referees for their helpful comments.

REFERENCES

[1] Akavia, A. (2008) "Learning Noisy Characters, Multiplication Codes and Hardcore Predicates." Ph.D. Thesis, Massachusetts Institute of Technology.

[2] Akavia, A. (2009) "Solving Hidden Number Problem with One Bit Oracle and Advice," in Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 337–354. Springer, Heidelberg

[3] Akavia, A., Goldwasser, S., and Safra, S. (2003) "Proving Hard-Core Predicates Using List Decoding," in FOCS 2003, pp. 146–157. IEEE Computer Society, Washington, DC.

[4] Alexi, W., Chor, B., Goldreich, O., and Schnorr, C.P. (1988) "RSA and Rabin Functions: Certain Parts are as Hard as the Whole," in SIAM Journal on Computing, **17(2)**, 194–209.

[5] Ben-Or, M., Chor, B., and Shamir, A. (1983) "On the Cryptographic Security of Single RSA Bits," Johnson, D.S., Fagin, R., Fredman, M.L., Harel, D., Karp, R.M., Lynch, N.A., Papadimitriou, C.H., Rivest, R.L., Ruzzo, W.L., Seiferas, J.I. (eds.) STOC 1983, pp. 421–430. ACM, New York.

[6] Boneh, D., and Shparlinski, I. (2001) "On the Unpredictability of Bits of the Elliptic Curve Diffie–Hellman Scheme," in Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 201–212. Springer, Heidelberg.

[7] Boneh, D., and Venkatesan, R. (1996) "Hardness of Computing the Most Significant Bits of Secret Keys in Diffie-Hellman and Related Schemes," in Koblitz, N. (ed.) CRYPTO 1996. LNCS, vol. 1109, pp. 129–142. Springer, Heidelberg.

[8] Boneh, D., and Venkatesan, R. (1997) "Rounding in Lattices and its Cryptographic Applications," in Saks, M.E. (ed.) SODA 1997, pp. 675–681. ACM/SIAM, Philadelphia.

[9] Duc, A., and Jetchev, D. (2012) "Hardness of Computing Individual Bits for One-Way Functions on Elliptic Curves," in Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 832–849. Springer, Heidelberg.

[10] De Mulder, E., Hutter, M., Marson, M.E., and Pearson, P. (2013) "Using Bleichenbacher's Solution to the Hidden Number Problem to Attack Nonce Leaks in 384-Bit ECDSA," in Bertoni, G., Coron, J.S. (eds.) CHES 2013. LNCS, vol. 8086, pp. 435–452. Springer, Heidelberg.

[11] Verheul, E.R. (2000) "Certificates of Recoverability with Scalable Recovery Agent Security," in Imai, H., Zheng, Y. (eds.) PKC 2000. LNCS, vol. 1751, pp. 258–275. Springer, Heidelberg.

[12] Fazio, N., Gennaro, R., Perera I.M., and Skeith III, W.E. (2013) "Hard-Core Predicates for a Diffie-Hellman Problem over Finite Fields," in Canetti, R., Garay, J.A. (eds.) CRYPTO 2013. LNCS, vol. 8043, pp. 148–165. Springer, Heidelberg.

[13] Goldreich, O., and Levin, L.A. (1989) "A Hard-Core Predicate for all One-Way Functions," in Johnson, D.S. (ed.) STOC 1989, pp. 25–32. ACM, New York.

[14] Aranha, D.F., Fouque, P.-A., Gérard B., Kammerer, J.-G., Tibouchi., M, and Zapalowicz, J.-C. (2014) "GLV/GLS Decomposition, Power Analysis, and Attacks on ECDSA Signatures with Single-Bit Nonce Bias," in Sarkar, P., Iwata, T. (eds.) ASIACRYPT 2014. LNCS, vol. 8873, pp. 262–281. Springer, Heidelberg.

[15] Kushilevitz, E., and Mansour, Y. (1991) "Learning Decision Trees Using the Fourier Sprectrum," in Koutsougeras, C., Vitter, J.S. (eds.) STOC 1991, pp. 455–464. ACM, New York.

[16] Morillo, P., and Ràfols, C. (2009) "The Security of All Bits Using List Decoding," in Jarecki, S., Tsudik, G. (eds.) PKC 2009. LNCS, vol. 5443, pp. 15–33. Springer, Heidelberg.

[17] Mansour, Y. (1994) "Learning Boolean Functions via the Fourier Transform," in Roychowdhury, V., Siu, K.Y., Orlitsky, A. (eds.) Theoretical Advances in Neural Computation and Learning, pp. 391–424. Kluwer Academic Publishers.

[18] Shparlinski, I. (2002) "Playing "Hide-and-Seek" in Finite Fields: Hidden Number Problem and its Applications," in Proceedings of the Seventh Spanish Meeting on Cryptology and Information Security, vol. 1, pp. 49–72. University of Oviedo.

[19] Shparlinski, I., and Winterhof, A. (2004) "A Nonuniform Algorithm for the Hidden Number Problem in Subgroups," in Bao, F., Deng, R.H., Zhou, J. (eds.) PKC 2004. LNCS, vol. 2947, pp. 416–424. Springer, Heidelberg.

[20] Wang, M., Zhan, T., and Zhang, H. (2014) "Bits Security of the CDH Problems over Finite Fields," in Cryptology ePrint Archive, Report 2014/685. http://eprint.iacr.org/2014/685.