On the Hidden Shifted Power Problem

Igor Shparlinski

Macquarie University

Joint work with:

Jean Bourgain, Moubariz Garaev and Sergei Konyagin

Introduction

Set-up and Motivation

Let $\mathbb{F}_q$ be a finite field of q elements.

For $e \mid q-1$ with $e \le (q-1)/2$ and $s \in \mathbb{F}_q$, $\mathcal{O}_{e,s}$ denote oracle that on every input $x \in \mathbb{F}_q$ outputs $\mathcal{O}_{e,s}(x) = (x+s)^e$ for some "hidden" $s \in \mathbb{F}_q$:

$$x \to \mathcal{O}_{e,s} \to (x+s)^e$$

Hidden Shifted Power Problem:

> **HSPP:** given $\mathcal{O}_{e,s}$ for some **unknown** $s \in \mathbb{F}_q$, find s

We also consider the following two versions of the *Shifted Power Identity Testing*:

> **SPIT-1:** given $\mathcal{O}_{e,s}$ for some **unknown** $s \in \mathbb{F}_q$ and **known** $t \in \mathbb{F}_q$, decide whether $s = t$ provided that the call $x = -t$ is forbidden

and

> **SPIT-2:** given $\mathcal{O}_{e,s}$ and $\mathcal{O}_{e,t}$ for some **unknown** $s, t \in \mathbb{F}_q$ decide whether $s = t$.

Side Remark

These problems are special cases of the "black-box" polynomial interpolation and identity testing for arbitrary polynomials given by **straight-line programs**: an instruction what operations to execute in order to evaluate $f(x)$

Example: Evaluating the polynomial

$$f(X) = (X - 3)(X + 2)^{100} + X^{200}$$

1. Read x
2. Add 2 to x
3. Rise (2) to the power 100
4. Subtract 3 from x
5. Multiply the results of (3) and (4)
6. Rise x to the power 200
7. Add the results of (5) and (6)
8. Output (7)

Complicated polynomials may have very short straight-line programs.

Classical Example: **determinant**

Classical Problem: show that **permanent** does not have a short straight-line program.

Straigh-Line Program Testsing:
Given two straight-line programs for multivariate polynomials f and g decide whether $f = g$ (as polynomials or functions over some fixed field).

The area has a long history in theoretic computer science and cryptography.

Here we consider very special polynomials given by straight line programs of length 2.

Observation

Observation: Returning the values of $(x+s)^e$

$$\Updownarrow$$

Giving u such that $x+s = u \times \mu$ for some $\mu \in \mathbb{F}_q$ with $\mu^e = 1$

$$\Updownarrow$$

returning the values of $\chi(x+s)$ for some fixed multiplicative character χ of $\mathbb{F}_q^*$.

In this form:

van Dam & Hallgren & Ip, **2006**:
an efficient quantum algorithm in the case of a *quantum* oracle $\mathcal{O}_{e,s}$ (that is, an oracle which can talk to a quantum computer).

Vercauteren, **2008**:
The same question under the name of *Hidden Root Problem* in relation to the **fault attack** on *pairing based protocols on elliptic curves*.

Boneh & Lipton; Damgård; Peralta, **1990-2000**:
Links between **HSPP** with $e = (p-1)/2$ (i.e., with the Legendre symbol) and cryptography, e.g. hashing.

Efficiency Meassures

- Number of Oracle Calls
(in cryptographic applications "calls" are expensive, they are induced hardware faults)
- Running Time

Two Straightforward Solutions

- **HSPP:** query $\mathcal{O}_{e,s}$ on $e+1$ arbitrary elements $x \in \mathbb{F}_q$ and then interpolate the results:

$$\text{Oracle Calls} = e \qquad \text{Time} = e(\log q)^{O(1)}$$

- **SPIT-1,2:** query $\mathcal{O}_{e,s}$ and $\mathcal{O}_{e,t}$ on N random elements $x \in \mathbb{F}_q$ and compare the results:

$$\text{Oracle Calls} = N \qquad \text{Time} = N(\log q)^{O(1)}$$

$$\text{Success Prob.:} = \left(1 - \frac{e}{p}\right)^N \le 2^{-N}$$

We will measure our progress (. . . and failures) against these naive solutions.

We concentrate on the case of a prime $q = p$.

Some of our results are compact and *nicely looking*, some are rather technical and *ugly* . . . but they do the job, lead to better algorithms.

You will see examples of both types.

Our Results

HSPP

HSPP: Small e

Let $e \mid p-1$ with $e \le p^{1-\delta}$.

Deterministic algorithm

For any $\varepsilon > 0$, it finds s in

- Calls $= O(1)$, Time $= e^{1+\varepsilon}(\log p)^{O(1)}$, provided we are given ℓ-th power nonresidues for all primes $\ell \mid e$ (or the ERH holds)
- Calls $= O(1)$, Time $= ep^{\varepsilon}$

Probabilistic algorithm

It finds s in expected number of

Calls $= O(\log p / \log(p/e))$ and Time $= ep^{o(1)}$

HSPP: Large e

<u>Deterministic algorithm</u>

For any $\varepsilon > 0$ it finds s in

- Calls $= O(\log p / \log(p/e))$, Time $= p(\log p)^{O(1)}$
- Calls $= O(\log p / \log(p/e))$, Time $= e^{1+\varepsilon}(\log p)^{O(1)}$, provided we are given ℓ-th power nonresidues for all primes $\ell \mid e$ (or the ERH holds)

<u>Note:</u> If $e \le p^{1-\delta}$ for some $\delta > 0$ then

$$\log p / \log(p/e) = O(1).$$

SPIT

SPIT-1: (that is, t is known)

Let $e \mid p-1$ and let we are given an oracle $\mathcal{O}_{e,s}$.

Deterministic algorithm

It tests $s = t$:

- For any $e \le (p-1)/2$, in
$$\text{Time} = e^{1/4} p^{o(1)}$$
- For $e \le p^{\delta}$, in
$$\text{Time} = e^{c_0 \delta} (\log p)^{O(1)},$$
where c_0 is a constant.

The constant c_0 can be explicitly evaluated, but we have never done so.

SPIT-2: (that is, t is unknown)

Let $e \mid p-1$ and let we are given oracles $\mathcal{O}_{e,s}$ and $\mathcal{O}_{e,t}$.

Deterministic algorithm

- For any $e \le (p-1)/2$,
$$\text{Time} = p^{1/2+o(1)}$$
- For any $e \le (p-1)/2$,
$$\text{Time} = \max\{e^{1/2}p^{o(1)}, e^2 p^{-1+o(1)}\}.$$
- For $e \le p^{\delta}$,
$$\text{Time} = e^{C_0 \delta^{1/3}} (\log p)^{O(1)}$$

The constant C_0 can be explicitly evaluated, but we have never done so.

Methods and Algorithms

HSPP

(i) Query $\mathcal{O}_{e,s}$ at several values of j, e.g. $j = 1, \ldots, m$ for some small m, getting $A_j = (s + j)^e$.

(ii) Find sets $\mathcal{S}_j$ of solutions to $A_j = u^e$, note that $s \in \mathcal{S}_j - j$.

(iii) Find their intersection

$$\mathcal{S} = \bigcap_{j \in [1,m]} (\mathcal{S}_m - j)$$

(iv) Prove that for m not too large, $\#\mathcal{S}$ is small.

(v) Query $\mathcal{O}_{e,s}$ for all $x \in -\mathcal{S}$ until it returns 0.

Step **(ii)**: It is well studied problem of root extraction in finite fields. Unfortunately still there is no polynomial time **deterministic** algorithm (even for $e = 2$) unless we are given ℓ-th power nonresidues for all primes $\ell \mid e$ (or the ERH holds).

Sometimes we can circumvent this problem but sometimes it holds us back (and so we request these non-residues to be given).

Step **(iii)**: we do not know how to do this more efficiently that directly from the definition. . .

Step **(iv)** is the key point in our approach.

The sets $\mathcal{S}_x$ are shifted co-sets of the multiplicative group

$$\mathcal{G}_e = \{\mu \in \mathbb{F}_q \ : \ \mu^e = 1\}$$

of residues of order e.

So, the problem has a natural multiplicative structure associate with it.

$$\Downarrow$$

We use some new results about the intersections of shifted co-sets and also some old and new number theoretic estimates of multiplicative character sums.

Technical Tools

Preliminary Shrinking the Search Set $\mathcal{S}$

Heath-Brown & Konyagin, **1999**: $m = 1$
Shkredov & Vyugin, **2011**: any $m \geq 1$

Lemma 1 *Assume that for an integer $m \geq 1$,*

$$p \geq 3me^{1+1/(2m+1)}.$$

Then for pairwise distinct $\mu_1, \ldots, \mu_m \in \mathbb{F}_p^*$ and arbitrary $\lambda_1, \ldots, \lambda_m \in \mathbb{F}_p^*$ the bound

$$\#\left(\mathcal{G}_e \cap (\lambda_1 \mathcal{G}_e + \mu_1) \cap \ldots \cap (\lambda_m \mathcal{G}_e + \mu_m)\right) \ll e^{\frac{m+1}{2m+1}}$$

holds, where the implied constant depends on m.

Note: The RHS of Lemma 1 approaches $e^{1/2}$ when m increases.

So, for any ε in $O(1)$ steps at the Step **(iii)** we obtain a set of size $e^{1/2+\varepsilon}$.

Further Shrinking the Search Set $\mathcal{S}$

We derive and use new bounds of multiplicative characters sums that stems from a series of results of

Karatsuba, **1992**:
Friedlander & Iwaniec, **1993**:
Chang, **2009**:

The aim is to get an improvement of the general bound

$$\left|\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}}\chi(x+y)\right| \le \sqrt{p\#\mathcal{X}\#\mathcal{Y}}$$

that holds for arbitrary sets $\mathcal{X}, \mathcal{Y} \subseteq \mathbb{F}_p$.

In our and all other works one of the sets is always assumed to be "structured" (e.g. an interval or d-spaced).

Improvements??

Each of these Steps **(ii)–(iv)** can be the bottleneck, depending on the value of e.

Step **(iii)**: We do not know any nontrivial algorithm for finding the set intersection.

Question: Any quantum speed-up?

Not the that the oracle $\mathcal{O}_{e,s}$ is *classical* here.

SPIT-1,2

As before, let $\mathcal{G}_e \subseteq \mathbb{F}_q^*$ be the multiplicative group of order $e \mid q-1$, that is,

$$\mathcal{G}_e = \{\mu \in \mathbb{F}_q \ : \ \mu^e = 1\}.$$

We write,

$$F_{s,t}(X) = \prod_{\mu \in \mathcal{G}_e} (X + s - \mu(X+t))\,.$$

Our approach is based on the idea of choosing a small "test" set $\mathcal{X}$, which nevertheless is guaranteed to contain at least one non-zero of the polynomial $F_{s,t}$ for any $s \neq t$.

This is based on a careful examination of the roots of $F_{s,t}$ and relating it to some classical number theoretic problems about the distribution of elements of small subgroups of finite fields.

Technical Tools

Bounding the Number of Solutions of Some Congruences

Ayyad, Cochrane and Zheng, **1996**:
Cilleruelo & Garaev, **2010**:, *Garaev & Garcia,* **2008**:

Lemma 2 *Uniformly over integers a and H, the congruence*

$$(a + x_1)(a + x_2) \equiv (a + x_3)(a + x_4) \pmod p,$$
$$1 \le x_1, x_2, x_3, x_4 \le H,$$

has $H^4/p + O(H^{2+o(1)})$ solutions as $H \to \infty$.

Additive combinatorics in algebraic number fields

Analogue of *Bourgain, Konyagin & Shaprlinski,* **2008**: ($\mathbb{K} = \mathbb{Q}$)
Another approach: *Cilleruelo, Ramana & Ramaré,* **2010**:

Lemma 3 *Let* $\mathcal{A}, \mathcal{B} \subseteq \mathbb{K}$, *where* $d = [\mathbb{K} : \mathbb{Q}]$ *be finite sets with elements of logarithmic height at most* H. *For some* $c(d)$, *depending only on* d,

$$\#(\mathcal{A}\mathcal{B}) > \exp\left(-c(d)\frac{H}{\sqrt{\log H}}\right)\#\mathcal{A}\#\mathcal{B}.$$

Effective Hilbert's *Nullstellensatz*

From *Krick, Pardo & Sombra,* **2001**: we derive (where we care only about the size of b and do not need to estimate other parameters):

Lemma 4 *Let $P_1, \ldots, P_N, f \in \mathbb{Z}[Z_1, \ldots, Z_n]$ be $N+1 \geq 2$ polynomials in n variables of degree at most $D \geq 3$ and of logarithmic height at most H such that f vanishes on the variety*

$$P_1(Z_1, \ldots, Z_n) = \ldots = P_N(Z_1, \ldots, Z_n) = 0.$$

There are positive integers b and r with

$$\log b \leq C(n)D^{n+1}(H + \log N + D)$$

and polynomials $Q_1, \ldots, Q_N \in \mathbb{Z}[Z_1, \ldots, Z_n]$ such that

$$P_1Q_1 + \ldots + P_NQ_N = bf^r,$$

where $C(n)$ depends only on n.

In our case, $n = 2$, not no better bound seems to be known.

Finite Fields

For $\mathcal{A} \subseteq \mathbb{F}_q$, let $\mathcal{A}^{(\nu)}$ be the ν-fold product set

$$\mathcal{A}^{(\nu)} = \{a_1 \ldots a_\nu \ : \ a_1 \ldots a_\nu \in \mathcal{A}\}$$

Lemma 5 *Let $\nu \geq 2$ be a fixed integer. Assume that*

$$h < p^{1/(\nu^2-1)}.$$

For $s \in \mathbb{F}_p$ we consider the set

$$\mathcal{A} = \{x + s \ : \ 1 \leq x \leq h\} \subseteq \mathbb{F}_p.$$

Then

$$\#(\mathcal{A}^{(\nu)}) > h^{\nu + o(1)}.$$

Note: The bound is tight as

$$\#(\mathcal{A}^{(\nu)}) \leq (\#\mathcal{A})^\nu \leq h^\nu$$

Interpretation: Intervals generate very large subgroups of $\mathbb{F}_p^*$.

Lemma 6 *Fix $\nu \ge 1$. Assume that*

$$h < p^{c\nu^{-4}},$$

where c is a certain absolute constant. For pair-wise distinct $s, t \in \mathbb{F}_p$ we consider the set

$$\mathcal{A} = \left\{ \frac{x+s}{x+t} \ : \ 1 \le x \le h \right\} \subseteq \mathbb{F}_p.$$

Then

$$\#(\mathcal{A}^{(\nu)}) > h^{\nu+o(1)}.$$

Note: The bound is tight as

$$\#(\mathcal{A}^{(\nu)}) \le (\#\mathcal{A})^{\nu} =\le h^{\nu}$$

Interpretation: Values of linear-fraction functions on intervals generate very large subgroups of $\mathbb{F}_p^*$.

SPIT-1: (that is, t is known)

Idea

Clearly, if

$$\mathcal{O}_{e,s}(x) = \mathcal{O}_{e,t}(x)$$

for some $x \in \mathbb{F}_q^*$ then $F_{s,t}(x) = 0$ or

$$\frac{x+s}{x+t} \in \mathcal{G}_e \tag{1}$$

(provided $x + t \neq 0$). We now choose

$$\mathcal{X} = \{y^{-1} - t \ : \ y \in \mathcal{Y}\} \tag{2}$$

for some set $\mathcal{Y} \subseteq \mathbb{F}_q^*$. Then the condition (1) means that a shift of $\mathcal{Y}$ is contained inside of a coset of $\mathcal{G}_e$, that is, with $r = (s-t)^{-1}$, we have

$$\mathcal{Y} + r \subseteq r\mathcal{G}_e \tag{3}$$

<u>Goal:</u> find a "small" set $\mathcal{Y} \subseteq \mathbb{F}_q^*$ such that its shifts cannot be inside of any coset of $\mathcal{G}_e$ (we note that r is unknown).
<u>Idea:</u> Choose $\mathcal{Y}$ as a short interval of h consecutive integers and define $\mathcal{X}$ by (2).

Algorithm for small e

Immediate from Lemma 5:

Products of sufficiently many copies of an interval cannot be locked in a co-sets of a small small subgroup.

If e is small, take $h = \left\lceil e^{c_0 \delta} \right\rceil$ for a sufficiently large c_0 and $\mathcal{Y} = [1, h]$ and see that (3) is impossible:

For a *known* $t \in \mathbb{F}_p$ and $e \le p^\delta$, we decided whether $s = t$ in

$$\text{Time} = e^{c_0 \delta} (\log p)^{O(1)},$$

where c_0 is a constant.

If $e = p^{o(1)}$ then $\text{Time} = e^{o(1)} (\log p)^{O(1)}$.

Large e: $e^{1/4}$ Algorithm

Consider $\mathcal{I} = [a+1, a+H] \subseteq [0, p-1]$ of size $H < p^{1/3}$.

Fix some integer $m \geq 1$ so that p and e satisfy the condition of Lemma 1.

Set

$$\ell = m!, \quad \ell_\nu = m!/(\nu{+}1),\ \nu = 1, \ldots, m{-}1, \quad K = \lfloor H/\ell \rfloor .$$

Let $\mathcal{J} = \{a+\ell, \ldots, a+\ell K\}$. Thus $\mathcal{J} \subseteq \mathcal{I}$. Let $\mathcal{A} = \mathcal{J}/\mathcal{J}$, that is,

$$\mathcal{A} = \{j_1/j_2 \ : \ j_1, j_2 \in \mathcal{J}\} \subseteq \mathbb{F}_p.$$

Now, let $N(\alpha)$ be the number of solutions to

$$\frac{a+\ell h}{a+\ell i} = \alpha \quad i,h \in [1,K],$$

Clearly $N(\alpha) > 0 \quad \Leftrightarrow \quad \alpha \in \mathcal{A}$.

Furthermore

$$\sum_{\alpha \in \mathcal{A}} N(\alpha)^2 = T,$$

where T is the number of solutions to:

$$\frac{a+\ell h}{a+\ell i} = \frac{a+\ell j}{a+\ell k}, \quad i,j,h,k \in [1,K].$$

or to

$$(a+\ell i)(a+\ell j) = (a+\ell h)(a+\ell k), \quad i,j,h,k \in [1,K].$$

By Lemma 2 we see that

$$\sum_{\alpha\in\mathcal{A}} N(\alpha)^2 \le H^{2+o(1)}$$

Also, we have the trivial relation

$$\sum_{\alpha\in\mathcal{A}} N(\alpha) = K^2$$

Therefore, by the Cauchy inequality

$$K^4 = \left(\sum_{\alpha\in\mathcal{A}} N(\alpha)\right)^2 \le \#\mathcal{A} \sum_{\alpha\in\mathcal{A}} N(\alpha)^2 \le \#\mathcal{A} K^{2+o(1)}$$

Hence $\#\mathcal{A}$ is large:

$$\#\mathcal{A} \ge K^{2+o(1)} = H^{2+o(1)}. \tag{4}$$

Next we observe that

$$\mathcal{A} + \nu \subseteq \{(\nu+1)u \ : \ u \in \mathcal{I}/\mathcal{I}\},$$

since

$$\frac{a+\ell h}{a+\ell i} + \nu = (\nu+1)\frac{a+\nu\ell_\nu i+\ell_\nu h}{a+\ell i}.$$

and

$$\nu\ell_\nu i + \ell_\nu h \le (\nu+1)\ell_\nu K \le H.$$

Clearly if $\mathcal{I} \in r\mathcal{G}_e$ then $\mathcal{A} \subseteq \mathcal{G}_e$ and $\mathcal{A}+\nu \subseteq (\nu+1)\mathcal{G}_e$. The system of equations

$$x_0 + \nu = x_\nu, \quad x_\nu \in (\nu+1)\mathcal{G}_e, \quad \nu = 0, \dots, m-1,$$

has at least $\#\mathcal{A}$ solutions of the form $x_0 \in \mathcal{A}$, $x_\nu = x_0 + \nu$, $\nu = 1, \dots, m$.

By Lemma 1 (bound on the intersection of m shifted co-sets of $\mathcal{G}_e$), we have

$$\#\mathcal{A} \ll e^{(m+1)/(2m+1)} \tag{5}$$

We see that for

$$H = \left\lfloor e^{1/4+\varepsilon} \right\rfloor$$

for some $\varepsilon > 0$. For a sufficiently large m we see that (4) and (5) are incomparable.

Choosing $\mathcal{Y} = [1, H]$ and recalling (3), we now complete the proof.

SPIT-2: (that is, t is unknown)

Idea

We cannot use

$$\mathcal{X} = \{y^{-1} - t \ : \ y \in \mathcal{Y}\}$$

anymore and have to work with

$$\frac{x+s}{x+t} \in \mathcal{G}_e \tag{6}$$

directly.

Goal: Find a "small" set $\mathcal{X} \subseteq \mathbb{F}_q^*$ such that the ν-fold product set of $(x+s)/(x+t)$, $x \in \mathcal{X}$ is large. Then (6) cannot hold unless $\boxed{s = t}$.

Idea: Choose $\mathcal{X}$ as a short interval of h consecutive integers, and test (6) by comparing $O_{e,s}(x)$ and $O_{e,t}(x)$ for $x \in \mathcal{X}$.

Algorithm for small e

Immediate from Lemma 6 (ν-fold product set of

$$\mathcal{A} = \left\{ \frac{x+s}{x+t} \ : \ 1 \le x \le h \right\} \subseteq \mathbb{F}_p.$$

is large).

If $e \le p^{\delta}$ is small, take $h = \left\lceil e^{c_0 \delta^{1/3}} \right\rceil$ for a sufficiently large c_0 and $\mathcal{Y} = [1, h]$ and see that (6) is impossible:

For a *unknown* $t \in \mathbb{F}_p$ and $e \le p^{\delta}$, we decided whether $s = t$ in

$$\mathsf{Time} = e^{c_0 \delta^{1/3}} (\log p)^{O(1)},$$

where c_0 is a constant.

If $e = p^{o(1)}$ then $\mathsf{Time} = e^{o(1)} (\log p)^{O(1)}$.

Algorithm for large e

Lemma 2: multiplicities of residues of $(x+u)(y+u)$

$$\Downarrow$$

For any interval $\mathcal{I} = [r+1, r+h] \subseteq \mathbb{F}_p$ the products uv, $u, v \in \mathcal{I}$, take a lot of distinct values:

$$\#\{uv \ : \ u, v \in \mathcal{I}\} \gg \min\{H^{1/2}p^{1/2}, H^{2+o(1)}\}$$

$$\Downarrow$$

The interval $\mathcal{I}$ is not contained in a small subgroup.

The classical *Burgess and Weil bounds* also work in some ranges.

Other Applications

Congruences

The following result in the case $\nu = 4$ solves an open problem *Cilleruelo & Garaev,* **2010**:.

Let $\nu \geq 2$ be a fixed integer, $\lambda \not\equiv 0 \pmod p$. Assume that for some sufficiently large positive integer h and prime p we have

$$h < p^{1/(\nu^2-1)}.$$

Then for any $s \in \mathbb{F}_p$ for the number $J_\nu(\lambda; h)$ of solutions of the congruence

$$(x_1+s)\ldots(x_\nu+s) \equiv \lambda \pmod p, \qquad 1 \leq x_1, \ldots, x_\nu \leq h,$$

we have the bound

$$J_\nu(\lambda; h) < \exp\left(c(\nu)\frac{\log h}{\log\log h}\right),$$

where $c(\nu)$ depends only on ν.

Polynomial Factorisation

The following algorithm (still in progress!!) improves the result of *Shoup*, **1991**:

There is a deterministic algorithm that, given a squarefree polynomial $f \in \mathbb{F}_p[X]$ of degree $n = p^\alpha$ that fully splits over $\mathbb{F}_p$, finds in time $p^{\vartheta + o(1)}$ a factor $g \mid f$ of degree $1 \le \deg g < n$ where

$$\vartheta = \begin{cases} 1/2, & \text{if } \alpha \ge 1/2, \\ \dfrac{3 + \alpha - \sqrt{1 - 2\alpha + 9\alpha^2}}{4}, & \text{if } 1/2 > \alpha \ge \alpha_0, \\ \dfrac{80 - 119\alpha^2}{160 - 119\alpha}, & \text{if } \alpha < \alpha_0, \end{cases}$$

where

$$\alpha_0 = \frac{3280}{14399} = 0.22779\ldots.$$

Open Questions

- What about arbitrary fields?

  ...most of our tools do not work there, but some modifications are possible

- Find other applications of these methods?

- Better results for almost all p?

- More complicated polynomials? For example, $a(X+s)^e + b(X+t)^f$ or $f(X)^e$

- Can we do better with quantum algorithms?

  Given pairwise distinct $a_1, \ldots, a_\nu \in \mathbb{F}_p$ and arbitrary $x_1, \ldots, x_\nu \in \mathbb{F}_p$ how fast can we find the intersection of the solution sets to

  $$(u + x_i)^e = a_i, \quad i = 1, \ldots, \nu?$$

  Note that we know that this set is **small**, e.g.

  - $O\left(e^{1/2+o(1)}\right)$ is ν is large
  - $O\left(e^{2/3+o(1)}\right)$ if $\nu = 2$ (an interesting case too).

  It feels like a special case of the *Hidden Subgroup Problem* but with a classically given function f.