Playing "Hide-and-Seek" in Finite Fields: Hidden Number Problem and Its Applications

Igor E. Shparlinski

Centre for Advanced Computing: Algorithms and Cryptography

Macquarie University

igor@comp.mq.edu.au

Introduction

We describe a rather surprising, yet powerful, combination of

- **exponential sums**
- **lattice reduction algorithms**.

This combination has led to a number of cryptographic applications, helping to make rigorous several heuristic approaches.

It provides a two edge sword to:

- prove important **security** results;
- create powerful **attacks**

Examples:

- Bit security of the
  - Diffie–Hellman key exchange system,
  - Shamir message passing scheme,
  - XTR cryptosystem,
  - Rivest–Shamir–Wagner timed-release crypto.

- Attacks on the
  - Digital Signature Scheme (DSA),
  - Nyberg–Rueppel Signature Scheme.

Notation

p = prime number

$\mathbb{F}_p$ = finite field of p elements.

$\lfloor s \rfloor_m$ = the remainder of s on division by m.

For $\ell > 0$, $\mathrm{MSB}_{\ell,p}(x)$ denotes any integer u such that

$$||\lfloor x \rfloor_p - u| \le p/2^{\ell+1}.$$

$\mathrm{MSB}_{\ell,p}(x)$ $\approx$ ℓ most significant bits of x.

However this definition is more flexible.

In particular, ℓ **need not be an integer**.

Hidden Number Problem (HNP)

Boneh and Venkatesan, 1996

HNP: *Recover $\alpha \in \mathbb{F}_p$ such that for many known random $t \in \mathbb{F}_p$ we are given* $\mathrm{MSB}_{\ell,p}(\alpha t)$ *for some $\ell > 0$.*

B&V, 1996: a polynomial time algorithm to solve **HNP** with $\ell \approx \log^{1/2} p$.

The algorithm is based on **lattice reduction.**

Lattices

Let $\{\mathbf{b}_1, \ldots, \mathbf{b}_s\}$ be a set of linearly independent vectors in $\mathbb{R}^s$. The set of vectors

$$L = \{\mathbf{z} \mid \mathbf{z} = \sum_{i=1}^{s} c_i \mathbf{b}_i, \quad c_1, \ldots, c_s \in \mathbf{Z}\}$$

is called an s-dimensional full rank lattice. The set $\{\mathbf{b}_1, \ldots, \mathbf{b}_s\}$ is called a *basis* of L.

The closest vector problem

CVP: Given a vector $\mathbf{r} \in \mathbb{R}^s$ find a lattice vector $\mathbf{v} \in L$ with

$$\|\mathbf{r} - \mathbf{v}\| = \min_{\mathbf{z} \in L} \|\mathbf{r} - \mathbf{z}\|.$$

CVP is **NP**-complete.

Approximate solution?

Lenstra, Lenstra and Lovász, 1982
Kannan, 1987
Schnorr, 1987

Lemma 1 *There exists a deterministic polynomial time algorithm which, for a given lattice L and a vector $\mathbf{r} \in \mathbb{R}^s$, finds a lattice vector $\mathbf{v} \in L$ satisfying the inequality*

$$\|\mathbf{r} - \mathbf{v}\| \le \exp\left(C \frac{s \log^2 \log s}{\log s}\right) \min_{\mathbf{z} \in L} \|\mathbf{r} - \mathbf{z}\|.$$

for some absolute constant $C > 0$.

LLL: stretch factor $2^{s/2}$ (can be used as well)

Working with $2^{o(s)}$ is technically easier

HNP and CVP — B&V, 1996

Let $d \geq 1$ be integer. Given t_i, $u_i = \mathrm{MSB}_{\ell,p}(\alpha t_i)$, $i = 1, \ldots, d$, we build the lattice $\mathcal{L}(p, \ell, t_1, \ldots, t_d)$ spanned by the rows of the matrix:

$$\begin{pmatrix} p & 0 & \ldots & 0 & 0 \\ 0 & p & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & \vdots \\ 0 & 0 & \ldots & p & 0 \\ t_1 & t_2 & \ldots & t_d & 1/2^{\ell+1} \end{pmatrix}.$$

The **unknown** vector $\mathbf{v} = (\lfloor \alpha t_1 \rfloor_p, \ldots, \lfloor \alpha t_d \rfloor_p, \alpha/2^{\ell+1})$

- belongs to $\mathcal{L}(p, \ell, t_1, \ldots, t_d)$

- is close to the **known** vector $\mathbf{u} = (u_1, \ldots, u_d, 0)$:

$$\|\mathbf{v} - \mathbf{u}\| = O\left(p 2^{-\ell}\right).$$

Idea: Apply a CVP algorithm and *hope* that it will output $\mathbf{v}$.

How to make it rigorous?

We show that for almost all $t_1, \ldots, t_d$, $\mathbf{v}$ is the only lattice vector which can be so close to $\mathbf{u}$.

In fact, even within the approximation factor of Lemma 1, that is within the distance of order $p2^{-\ell+o(d)}$, this is still the **only** lattice vector.

Assume that $\mathbf{w} \equiv (\beta t_1, \ldots, \beta t_d, \beta/2^{\ell+1}) \pmod p$, with $\beta \not\equiv \alpha \pmod p$ is another lattice vector with

$$\|\mathbf{w} - \mathbf{u}\| \le p2^{-\ell+o(d)}.$$

Then

$$\|\mathbf{w} - \mathbf{v}\| \le p2^{-\ell+o(d)}. \qquad (1)$$

Therefore for each $i = 1, \ldots, d$

$$(\alpha - \beta)t_i \in [-p2^{-\ell+o(d)}, p2^{-\ell+o(d)}] \pmod p$$

For every fixed $\gamma \not\equiv 0 \pmod p$

$$\Pr_{t \in \mathbb{F}_p} (\gamma t \in [-h, h] \pmod p) \le \frac{2h+1}{p} \qquad (2)$$

Thus

$$\Pr_{t_1,\ldots,t_d \in \mathbb{F}_p} \left(\gamma t_i \in [-h, h] \pmod p,\ i = 1, \ldots, d\right) \le \left(\frac{2h+1}{p}\right)^d.$$

In our settings

$$\gamma = \alpha - \beta \qquad \text{and} \qquad h = p2^{-\ell+o(d)}.$$

Because β (and thus $\gamma = \alpha - \beta$) may belong to $p-1$ distinct residue classes we conclude that (1) holds with probability at most

$$P \le p\left(2^{-\ell+o(d)}\right)^d.$$

Choose $\ell = d = 2\left\lceil \log^{1/2} p \right\rceil$. Then

$$P \le \frac{1}{p}.$$

$$\boxed{\text{CVP algorithm returns } \mathbf{v} \text{ with prob. } \ge 1 - 1/p}$$

Extended HNP

HNP: *Recover $\alpha \in \mathbb{F}_p$ such that for many known random $t \in \mathbb{F}_p$ we are given* $\mathrm{MSB}_{\ell,p}(\alpha t)$ *for some $\ell > 0$.*

The condition that t is selected uniformly at random from $\mathbb{F}_p$ is too restrictive for applications.

Typically t is selected from a certain finite sequence $\mathcal{T}$ of elements of $\mathbb{F}_p$ which

- may have a nice and well-studied number theoretic structure (bit security of Diffie–Hellman key),
- may be rather "ugly" looking (attacks on DSA and Nyberg–Rueppel).

EHNP: *Recover $\alpha \in \mathbb{F}_p$ such that for many known random $t \in \mathcal{T}$ we are given* $\mathrm{MSB}_{\ell,p}(\alpha t)$ *for some $\ell > 0$.*

The same arguments as above apply to the **EHNP** ... but one needs an analogue of (2).

$$\Downarrow$$

$\mathcal{T}$ must have some **uniformity of distribution** properties.

Distribution of Sequences

Discrepancy $\mathcal{D}(\Gamma)$ of an N-element sequence $\Gamma = \{\gamma_1, \ldots, \gamma_N\}$ of elements of the interval $[0, 1]$ is defined as

$$\sup_{J \subseteq [0,1]} \left| \frac{A(J, N)}{N} - |J| \right|,$$

where $|J|$ is the length of the interval J and $A(J, N) = \# \{\gamma_n \in J,\ 1 \le n \le N\}$.

A finite sequence $\mathcal{T}$ of integers is Δ-*homogeneously distributed modulo* p (Δ-HD$_p$) if for any $a \in [1, p-1]$,

$$\{\lfloor at \rfloor_p / p\}, \qquad t \in \mathcal{T},$$

has the discrepancy at most Δ.

Putting Together

For a Δ-HD$_p$ sequence $\mathcal{T}$ instead of (2) we get

$$\Pr_{t \in \mathcal{T}} \left(\gamma t \in [-h, h] \pmod p \right) \le \frac{2h+1}{p} + \Delta.$$

<u>Nguyen&Shparlinski, 2000:</u>

Theorem 2 *Let* $\ell = \lceil \log^{1/2} p \rceil + \lceil \log \log p \rceil$ *and* $d = 2 \left\lceil \log^{1/2} p \right\rceil$*. Let* $\mathcal{T}$ *be* $2^{-\log^{1/2} p}$*-HD*$_p$*. There exists a deterministic polynomial time algorithm* $\mathcal{A}$ *such that for any fixed integer* $\alpha \in [0, p-1]$*, given* $2d$ *integers*

$$t_i \qquad \text{and} \qquad u_i = \mathrm{MSB}_{\ell,p}(\alpha t_i), \qquad i = 1, \ldots, d,$$

its output satisfies

$$\Pr_{t_1,\ldots,t_d \in \mathcal{T}} \left[\mathcal{A}(t_1, \ldots, t_d; u_1, \ldots, u_d) = \alpha \right] \ge 1 - 2^{-(\log p)^{1/2} \log \log p}$$

if $t_1, \ldots, t_d$ *are chosen uniformly and independently at random from the elements of* $\mathcal{T}$*.*

Discrepancy and Exponential Sums

Polya–Vinogradov, 1918:

$\mathcal{T}$ is Δ-HD$_p$ with

$$\Delta = O\left(\frac{\log p}{\#\mathcal{T}} \max_{1\le c\le p-1} \left|\sum_{t\in\mathcal{T}} \exp\left(2\pi i ct/p\right)\right|\right).$$

To use it we need an improvement up on the trivial bound

$$\left|\sum_{t\in\mathcal{T}} \exp\left(2\pi i ct/p\right)\right| \le \#\mathcal{T}$$

In many situatuions we have such resuslt which are quite enough . . . but what if only a very weak bound of the above esponential sums is know?

Using Very Weak Bounds

Shparlinski&Winterhof, 2003:

We can amplify it but considering k-sums

$$\{t_1 + \ldots + t_k \mid t_1, \ldots, t_k \in \mathcal{T}\}.$$

The discrepancy of this sequence:

$$\Delta_k = O\left(\frac{\log p}{\#\mathcal{T}} \max_{1 \le c \le p-1} \left|\sum_{t \in \mathcal{T}} \exp\left(2\pi i c t/p\right)\right|^k\right).$$

Any nontrvial saving γ against the trivial bound

$$\left|\sum_{t \in \mathcal{T}} \exp\left(2\pi i c t/p\right)\right| \le \gamma \#\mathcal{T}$$

will be risen to the kth power!

Important Example

Konyagin, 1992:

For any $1 > \varepsilon > 0$ there exists a constant $c(\varepsilon) > 0$ such that for any subgroup $\mathcal{G} \subseteq \mathbb{F}_p^*$ of order

$$T \geq \frac{\log p}{(\log \log p)^{1-\varepsilon}}$$

the bound

$$\max_{\gcd(\lambda,p)=1} \left| \sum_{r \in \mathcal{G}} \mathbf{e}_p(\lambda r) \right| \leq T \left(1 - \frac{c(\varepsilon)}{(\log p)^{1+\varepsilon}}\right)$$

holds.

Konyagin&Shparlinski, 1999:

For larger subgroups stronger bounds are known.

Modifications to the Algorithm

Chose

$$t_{11}, \dots, t_{1k}, \dots, t_{d1}, \dots, t_{dk} \in \mathcal{G}$$

and get integers u_{ij} with

$$\left| \left\lfloor \alpha r_{ij} \right\rfloor_p - u_{ij} \right| < p/2^{\ell+1}, \quad i = 1, \dots, d, \; j = 1, \dots, k.$$

For $i = 1, 2, \dots, d$ we put

$$v_i = \sum_{j=1}^{k} \left\lfloor \alpha r_{ij} \right\rfloor_p, \quad t_i = \left\lfloor \sum_{j=1}^{k} t_{ij} \right\rfloor_p, \quad u_i = \sum_{j=1}^{k} u_{ij}$$

The rest of the algorithm remains the same.

Good News: Bit Security of the Diffie–Hellman Key

Diffie–Hellman (DH) problem:

Given an element g of order τ modulo p, recover $K = \lfloor g^{xy} \rfloor_p$ from $\lfloor g^x \rfloor_p$ and $\lfloor g^y \rfloor_p$.

Typically, either $\tau = p - 1$ or $\tau = q$ – a large prime divisor of $p - 1$

The size of p and τ is determined by the present state of art in the **discrete logarithm problem**. Typically, p is about 500 bits, τ is at least 160 bits.

However after the common DH key $K = g^{xy}$ is established, only a small portion of bits of K will be used as a common key for some **private** key cryptosystem.

Assume that finding K is infeasible. Is it still infeasible to find certain bits of K?

Private Key |	Public Key

<u>Boneh&Venkatesan, 1996:</u>

for $\tau = p - 1$ (- small gap in the proof)

<u>González Vasco&Shparlinski, 2000:</u>

for "any" τ (+ fixing the gap in BV)

$\boxed{\text{YES!!!}}$

Assume we know how to recover ℓ most significant bits of $\lfloor g^{xy} \rfloor_p$ from from $X = \lfloor g^x \rfloor_p$ and $Y = \lfloor g^y \rfloor_p$.

Select a random $u \in [0, \tau - 1]$ and apply this algorithm to $X = \lfloor g^x \rfloor_p$ and $U = \lfloor Y g^u \rfloor_p = \left\lfloor g^{y+u} \right\rfloor_p$:

$$\mathrm{MSB}_{\ell,p}\left(g^{x(y+u)}\right) = \mathrm{MSB}_{\ell,p}\left(g^{xy} g^{xu}\right) = \mathrm{MSB}_{\ell,p}(\alpha t)$$

EHNP with $\alpha = g^{xy}$ and $t = g^{xu}$, $u \in [0, \tau - 1]$!!!

When γ^u is $2^{-\log^{1/2} p}$-HD$_p$? $(\gamma = g^x)$

<u>Shparlinski& Winterhof, 1999:</u>

Theorem 3 *For any* $\varepsilon > 0$ *there exists* $c > 0$ *such that for* $k = c \log^2 p$ *any* $\gamma \in \mathbb{F}_p$ *of order* $\tau \geq (\log p)^{1+\varepsilon}$ *the sequence*

$$\mathcal{T}_k = \{\gamma^{u_1} + \ldots + \gamma^{u_k},\ u_1, \ldots, u_k = 0, \ldots, \tau - 1\}$$

is $p^{-\delta}$*-HDp.*

If p is an n-bit prime and $\tau \geq (\log p)^{1+\varepsilon}$ then $\approx n^{1/2}$ most significant bits of the DH key are as secure as the whole key.

What Else?

Similar results for the **Shamir message passing scheme** (has not been worked out in details).

Shparlinski, 2000:
Li, Näslund, Shparlinski, 2002:

Similar results for the **XTR cryptosystem** of Lenstra&Verheul

Galbraith&Hopkins&Shparlinski, 2003:

Similar results for the **bilinear Diffie-Hellman bits**

In both case but for much large ordes.

Open Question: Extend the range.

Bad News: Attack on DSA

DSA: Proposed NIST, August 1991; US Federal Information Processing Standard 186, May 1994

<u>Public Data:</u>

q and p $=$ primes with $q|p-1$

$g \in \mathbb{F}_p$ $=$ a fixed element of order q.

$\mathcal{M}$ $=$ set of messages to be signed

$h : \mathcal{M} \to \mathbb{F}_q$ $=$ a hash-function.

The **secret key** is $\alpha \in \mathbb{F}_q^*$ which is known only to the **signer** (and publishes $A = \lfloor g^\alpha \rfloor_p$ – to be used for signature verification).

To sign a message $\mu \in \mathcal{M}$, the signer chooses a random integer $k \in \mathbb{F}_q^*$ usually called the *nonce*, and which must be kept **secret** and computes:

$$r(k) = \left\lfloor \left\lfloor g^k \right\rfloor_p \right\rfloor_q, \quad s(k, \mu) = \left\lfloor k^{-1} \left(h(\mu) + \alpha r(k) \right) \right\rfloor_q.$$

$(r(k), s(k, \mu))$ is the *DSA signature* of the message μ with a nonce k.

Assume that some bits of k are "leaked".

Howgrave-Graham&Smart, 1998:
Heuristic lattice based attack.

Nguyen, 1999:
Simpler and more powerful but still **heuristic** lattice based attack.

Nguyen&Shparlinski, 1999:
Rigorous lattice based attack.

Idea (Nguyen, 1999):

$$s(k,\mu) \equiv k^{-1}\left(h(\mu) + \alpha r(k)\right) \pmod q$$

$$\Downarrow$$

$$\alpha\, r(k)s(k,\mu)^{-1} \equiv k - h(\mu)s(k,\mu)^{-1} \pmod q.$$

If ℓ most significant bits of k are known then we know $\mathrm{MSB}_{\ell,q}\left(\alpha r(k)s(k,\mu)^{-1}\right)$.

EHNP with

$$t(k,\mu) = \left\lfloor r(k)s(k,\mu)^{-1}\right\rfloor_q, \quad (k,\mu) \in [1, q-1] \times \mathcal{M}.$$

Nguyen&Shparlinski, 1999:

$$W \;=\; \#\{h(\mu_1) = h(\mu_2), \quad \mu_1, \mu_2 \in \mathcal{M}\}.$$

$W/\#\mathcal{M}^2 \;=\;$ probability of *collision*.
Typically $W/|\mathcal{M}|^2 \approx q^{-1}$.

Theorem 4 *Let Q be a sufficiently large integer. The following statement holds with $\vartheta = 1/3$ for all primes $p \in [Q, 2Q]$, and with $\vartheta = 0$ for all primes $p \in [Q, 2Q]$ except at most $Q^{5/6+\varepsilon}$ of them. For any $\varepsilon > 0$ there exists $\delta > 0$ such that for any $g \in \mathbb{F}_p$ of order $q \geq p^{\vartheta+\varepsilon}$ the sequence*

$$t(k, \mu) = \left\lfloor r(k)s(k, \mu)^{-1} \right\rfloor_q, \quad (k, \mu) \in [1, q-1] \times \mathcal{M}.$$

is $q^{-\delta}$-HD$_q$, provided

$$W \leq \frac{\#\mathcal{M}^2}{q^{1-\delta}}.$$

Theoretically: If q is an n-bit prime and $\approx n^{1/2}$ most significant bits of k are known for $\approx n^{1/2}$ signatures then α can be recovered in polynomial time.

The proof uses:

- bounds of exponential sums with exponential functions (Konyagin&Shparlinski, 1999);
- **Weil's** bound;
- **Vinogradov's** method of estimates of double sums.

Main difficulty: The double reduction errases any number theoretric structure among the values of $r(k)$.

Practically: 4 bits of k are always enough, 3 bits are often enough, 2 bits are possibly enough as well.

Moral:

1. Do not use **small** k (to cut the cost of exponentiation in $r(k)$).

2. Protect your software/hardware against **timing/power attacks** when the attacker measures the time/power consumption and selects the signatures for which this value is smaller than "on average" – these signatures are likely to correspond to small k ($\sim$ faster exponentiation in $r(k)$).

3. Use quality **PRNG**'s to generate k, biased generators are dangerous.

4. Do not use **Arazi's cryptosystem** which combines DSA and Diffie-Hellman protocol – it leakes some bits of k (*Brown & Menezes*).

5. Do not buy CryptoLib from **AT&T**, it always uses odd values of k thus one bit is leaked immediately, one more and . . . .

Generalizations and Open Problems

Complete analogue of the bit security results for the DH key are also known ElGamal cryptosystem, Shamir message passing scheme and several others.

For **XTR** some non-trivial results are know as well (Li, Näslund, Shparlinski, 2002).

Attacks on other DSA-like schemes, including the **elliptic curve DSA**, of the same strength as on the original DSA (ElMahassni, Nguyen, Shparlinski, 2000–2001).

For the **Nyberg–Rueppel** scheme the range of p and q in which the results are nontrivial are narrower than in practical applications. Improve??? . . . Better bounds of exponential sums are required.