Public Key Cryptography Based on Partial Knowledge of Finite Fourier Transforms

Joseph H. Silverman

Brown University

Joint work with J. Hoffstein and J. Pipher (Brown University) and J. Schanck and W. Whyte (Security Innovations)

Colloquium, Microsoft Research, Redmond, WA

July 2013

Introduction

The shortest and closest vector problems (SVP, CVP) are interesting hard lattice problems.

There is a long history of cryptographic constructions based on problems that are equivalent to SVP and/or CVP in certain classes of lattices. For example:

1982 Merkle–Hellman: knapsack-based PKC

1997 Ajtai–Dwork: average case/worst case equivalence

1997 Goldwasser–Goldreich–Halevi: direct CVP-based PKC (very large key sizes)

1998 Hoffstein–Pipher–Silverman: NTRU via SVP and CVP in cyclic modular lattices (with practical key and ciphertext sizes)

And to the extent that I understand them, "learning with errors" (LWE) problems are (mostly?) reducible to SVP/CVP problems in certain lattices.

Outline

In 1999 a digital signature scheme called PASS was introduced by Hoffstein, Lieman, and Silverman. PASS is based on partial evaluation of Finite Fourier Transforms (FFT), which may be reduced to SVP/CVP in certain modular lattices. Unfortunately, PASS was vulnerable to transcript attacks.

In this talk I will describe a new PKC based on the partial-FFT problem, and a new variant of PASS that uses rejection sampling to eliminate transcript attacks.

OUTLINE

- Finite Fourier transforms and convolution products
- PASSEncrypt — A public key cryptostystem based on partial-FFT information
- PASSSign — A digital signature scheme based on partial-FFT information
- Partial FFT problems and lattice problems

Notation

We fix two public parameters:

N a prime number (say between 500 and 2000).
q a prime number satisfying $q \equiv 1 \pmod{N}$.

We write

$$\mathbb{F}_q = \mathbb{Z}/q\mathbb{Z}$$

for the finite field with q elements. The assumption on q and N means that we can find a primitive N'th root of unity w in $\mathbb{F}_q$. In other words,

$$w \in \mathbb{F}_q \text{ satisfies } w \neq 1 \text{ and } w^N = 1.$$

We work with N-dimensional vectors in $\mathbb{F}_q$,

$$\boldsymbol{a} = (a_0, \ldots, a_{N-1}) \in \mathbb{F}_q^N.$$

Finite Fourier Transforms

The **Finite Fourier Transform (FFT)** of a vector $\boldsymbol{a} \in \mathbb{F}_q^N$ is the vector

$$\mathcal{F}(\boldsymbol{a}) = \hat{\boldsymbol{a}} = (\hat{a}_0, \ldots, \hat{a}_{N-1})$$

whose k'th coefficient is given by the formula

$$\hat{a}_k = a_0 + a_1 w^k + a_2 w^{2k} + \cdots + a_{N-1} w^{(N-1)k}.$$

$\mathcal{F} : \mathbb{F}_q^N \to \mathbb{F}_q^N$ is a function mapping vectors to vectors.

The map $\mathcal{F}$ is a bijection. Its inverse is

$$\mathcal{F}^{-1}(\boldsymbol{b})_k = \frac{1}{N}\left(b_0 + b_1 w^{-k} + \cdots + b_{N-1} w^{-(N-1)k}\right).$$

Thus $\mathcal{F}$ is very far from being a one-way function.

Partial Knowledge of FFT

We partition the set of indices into two disjoint sets

$$\{0, 1, \ldots, N-1\} = S \cup T,$$

say with

$$s = \#S \approx N/2 \quad \text{and} \quad t = \#T \approx N/2,$$

and we define **Partial FFTs**

$$\mathcal{F}_S(\boldsymbol{a}) = (\hat{a}_i)_{i \in S} \quad \text{and} \quad \mathcal{F}_T(\boldsymbol{a}) = (\hat{a}_i)_{i \in T}.$$

The map $\mathcal{F}_S : \mathbb{F}_q^N \to \mathbb{F}_q^s$ does not have an easily computable inverse, but unfortunately that's because it doesn't have an inverse at all! It's not even close to being injective.

So we restrict the domain of $\mathcal{F}_S$ to a subset where it is (probably) injective.

Partial FFTs and Short Vectors

Let

$$\mathbb{T}^N = \{-1, 0, 1\}^N = \{\text{dim } N \text{ ternary vectors}\}.$$

We call elements of $\mathbb{T}^N$ **short vectors**. The map

$$\mathcal{F}_S : \mathbb{T}^N \longrightarrow \mathbb{F}_q^s$$

is almost certainly injective, but it is difficult to recover the ternary vector $\boldsymbol{a} \in \mathbb{T}^N$ from knowledge of only $\mathcal{F}_S(\boldsymbol{a})$. It is this hard problem that we exploit to create a public key cryptosystem and a digital signature scheme.

The **Short Vector Recovery from Partial Finite Fourier Transform Problem** (SVR-PFFT) is the problem of determining a ternary vector $\boldsymbol{a} \in \mathbb{T}^N$ from its partial-FFT $\mathcal{F}_S(\boldsymbol{a})$.

At the end I will explain how SVR-PFFT is naturally equivalent to a closest vector lattice problem.

An Obvious Ring Structure on $\mathbb{F}_q^N$

We add vectors in the usual way:

$$\boldsymbol{a} + \boldsymbol{b} = (a_0 + b_0, \ldots, a_{N-1} + b_{N-1}).$$

We can also multiply vectors in the obvious way, coordinate-by-coordinate:

$$\boldsymbol{a} \odot \boldsymbol{b} = (a_0 b_0, \ldots, a_{N-1} b_{N-1}).$$

The operations $+$ and $\odot$ make $\mathbb{F}_q^N$ into a ring, so for example,

$$\begin{aligned} (\boldsymbol{a} \odot \boldsymbol{b}) \odot \boldsymbol{c} &= \boldsymbol{a} \odot (\boldsymbol{b} \odot \boldsymbol{c}) && \text{Associative Law,} \\ \boldsymbol{a} \odot (\boldsymbol{b} + \boldsymbol{c}) &= \boldsymbol{a} \odot \boldsymbol{b} + \boldsymbol{a} \odot \boldsymbol{c} && \text{Distributive Law,} \\ &\vdots && \vdots \end{aligned}$$

Convolution Products

There is another kind of multiplication called **convolution product** whose definition is more complicated:

$$\boldsymbol{c} = \boldsymbol{a} \star \boldsymbol{b} \quad \text{with} \quad c_k = \sum_{i+j \equiv k \pmod N} a_i b_j.$$

Surprisingly, $+$ and $\star$ also make $\mathbb{F}_q^N$ into a ring, so $\star$ is associative, and $\star$ distributes over $+$.

Further, the FFT is a ring homomorphism between these two rings. In other words,

$$\begin{aligned} \mathcal{F}(\boldsymbol{a} + \boldsymbol{b}) &= \mathcal{F}(\boldsymbol{a}) + \mathcal{F}(\boldsymbol{b}), \\ \mathcal{F}(\boldsymbol{a} \star \boldsymbol{b}) &= \mathcal{F}(\boldsymbol{a}) \odot \mathcal{F}(\boldsymbol{b}). \end{aligned}$$

Thus $\mathcal{F}$ changes convolution product $\star$ into coordinate product $\odot$.

PASSEncrypt — A Partial-FFT PKC

System Parameters: N, q, and a small prime p (e.g., $p = 2$).

Key Creation:

$$\boldsymbol{f} \in \mathbb{T}^N \quad \text{private key is a small vector.}$$
$$\mathcal{F}_T(\boldsymbol{f}) \in \mathbb{F}_q^t \quad \text{public key is the } T\text{-partial-FFT of } \boldsymbol{f}.$$

Encryption:

$$\boldsymbol{m} \in \mathbb{T}^N \quad \text{(padded) plaintext is a small vector.}$$
$$\boldsymbol{r} \in \mathbb{T}^N \quad \text{nonce is a small vector.}$$

The ciphertext $(\boldsymbol{e}, \boldsymbol{e}', \boldsymbol{e}'')$ is three partial-FFTs:

$$\begin{aligned} \boldsymbol{e} &= \mathcal{F}_S(\boldsymbol{r}), \\ \boldsymbol{e}' &= \mathcal{F}_S(\boldsymbol{m}), \\ \boldsymbol{e}'' &= p\mathcal{F}_T(\boldsymbol{r}) \odot \mathcal{F}_T(\boldsymbol{f}) + \mathcal{F}_T(\boldsymbol{m}). \end{aligned}$$

PASSEncrypt (continued)

Decryption:

1. Use $\boldsymbol{f}$ to compute $\mathcal{F}_S(\boldsymbol{f})$.
2. Use $\boldsymbol{e} = \mathcal{F}_S(\boldsymbol{r})$ and $\boldsymbol{e}' = \mathcal{F}_S(\boldsymbol{m})$ to compute
$$p\boldsymbol{e} \odot \mathcal{F}_S(\boldsymbol{f}) + \boldsymbol{e}' = p\mathcal{F}_S(\boldsymbol{r}) \odot \mathcal{F}_S(\boldsymbol{f}) + \mathcal{F}_S(\boldsymbol{m}).$$
3. Use this and $\boldsymbol{e}''$ to recover the full FFT
$$p\mathcal{F}(\boldsymbol{r}) \odot \mathcal{F}(\boldsymbol{f}) + \mathcal{F}(\boldsymbol{m}) = \mathcal{F}(p\boldsymbol{r} \star \boldsymbol{f} + \boldsymbol{m}).$$
4. Apply $\mathcal{F}^{-1}$ to compute
$$p\boldsymbol{r} \star \boldsymbol{f} + \boldsymbol{m} \in \mathcal{F}_q^N.$$
5. Since $\boldsymbol{r}$, $\boldsymbol{f}$, and $\boldsymbol{m}$ are small vectors, choosing coordinates between 0 and $q-1$, we recover exactly the vector
$$p\boldsymbol{r} \star \boldsymbol{f} + \boldsymbol{m} \in \mathbb{Z}^N.$$
6. Reduce modulo p to recover the plaintext $\boldsymbol{m}$.

PASSEncrypt Summary

Public knowledge consists of:

- T-partial-FFT of the private key $\boldsymbol{f}$.
- S-partial-FFT of the (padded) plaintext $\boldsymbol{m}$.
- S-partial-FFT of the nonce $\boldsymbol{r}$.
- T-partial-FFT of the quantity $p\boldsymbol{r} \star \boldsymbol{f} + \boldsymbol{m}$.

Using $\boldsymbol{f}$, can compute the full FFT $\mathcal{F}(p\boldsymbol{r} \star \boldsymbol{f} + \boldsymbol{m})$.

The inverse FFT gives $p\boldsymbol{r} \star \boldsymbol{f} + \boldsymbol{m}$ in $\mathbb{F}_q^N$.

Smallness of $\boldsymbol{r}$, $\boldsymbol{f}$, and $\boldsymbol{m}$ gives $p\boldsymbol{r} \star \boldsymbol{f} + \boldsymbol{m}$ exactly.

Reducion mod p gives $\boldsymbol{m}$.

The hard problem is to recover any one of the small vectors $\boldsymbol{f}$, $\boldsymbol{m}$, $\boldsymbol{r}$ from their partial-FFTs.

Some Additional Notation

Before describing the digital signature scheme PASSSign based on partial-FFT, we need one more piece of notation.

The **infinity norm** of a vector $\boldsymbol{a} \in \mathbb{F}_q^N$ is the quantity

$$\|\boldsymbol{a}\| = \max\{|a_0|, \ldots, |a_{N-1}|\},$$

where we choose $-\frac{1}{2}q < a_i \leq \frac{1}{2}q$. We also let

$$\mathbb{T}^N(b) = \{\boldsymbol{a} \in \mathbb{T}^N : \text{at most } b \text{ nonzero coordinates}\}.$$

We note that if $\boldsymbol{f} \in \mathbb{T}^N$ and $\boldsymbol{c} \in \mathbb{T}^N(b)$, then

$$\|\boldsymbol{f} \star \boldsymbol{c}\| \leq b.$$

PASSSign: Preliminary Version

System Parameters: N, $q \equiv 1 \pmod{N}$, $w^N = 1$ in $\mathbb{F}_q$, $S \cup T = \{0, \ldots, N-1\}$, norm bound k, commitment bound b.

Key Creation:

$\boldsymbol{f} \in \mathbb{T}^N$ private key is a small vector.

$\mathcal{F}_T(\boldsymbol{f}) \in \mathbb{F}_q^t$ public key is the T-partial-FFT of $\boldsymbol{f}$.

Signing:

$\boldsymbol{m} \in \mathbb{T}^*$ document is a small vector.

$\boldsymbol{r} \in \mathbb{F}_q^N$ nonce is a vector with $\|\boldsymbol{r}\| \leq k$.

$\boldsymbol{c} \in \mathbb{T}^N(b)$ commitment equals $\mathsf{Hash}\big(\mathcal{F}_T(\boldsymbol{r}), \boldsymbol{m}\big)$.

$\boldsymbol{z} \in \mathbb{F}_q^N$ equals $\boldsymbol{f} \star \boldsymbol{c} + \boldsymbol{r}$.

The signature is the triple

$$(\boldsymbol{z}, \boldsymbol{c}, \boldsymbol{m}).$$

PASSSign: Preliminary Version (continued)

Verification:

1. If $\|\boldsymbol{z}\| > b + k$ or $\boldsymbol{c} \notin \mathbb{T}^N(b)$, return **Invalid**.
2. Compute

$$\boldsymbol{c}' = \mathsf{Hash}\big(\mathcal{F}_T(\boldsymbol{z}) - \mathcal{F}_T(\boldsymbol{f}) \odot \mathcal{F}_T(\boldsymbol{c}), \boldsymbol{m}\big).$$

3. If $\boldsymbol{c}' \neq \boldsymbol{c}$, return **Invalid**.
4. Return **Valid**.

Why It Works:

$$\begin{aligned}\boldsymbol{c}' &= \mathsf{Hash}\big(\mathcal{F}_T(\boldsymbol{z}) - \mathcal{F}_T(\boldsymbol{f}) \odot \mathcal{F}_T(\boldsymbol{c}), \boldsymbol{m}\big) \\ &= \mathsf{Hash}\big(\mathcal{F}_T(\boldsymbol{z} - \boldsymbol{f} \star \boldsymbol{c}), \boldsymbol{m}\big) \\ &= \mathsf{Hash}\big(\mathcal{F}_T(\boldsymbol{r}), \boldsymbol{m}\big) \\ &= \boldsymbol{c}. \quad \checkmark\end{aligned}$$

Why Is It Hard To Forge?

What does a forger need to do?

1. She starts with an arbitray vector $\boldsymbol{r} \in \mathbb{F}_q^t$.

2. Next she sets

$$\boldsymbol{c} = \mathsf{Hash}\big(\boldsymbol{r}, \boldsymbol{m}\big).$$

3. Finally she selects a vector $\boldsymbol{z} \in \mathbb{F}_q^N$ whose partial-FFT satisfies

$$\mathcal{F}_T(\boldsymbol{z}) = \mathcal{F}_T(\boldsymbol{r}) + \mathcal{F}_T(\boldsymbol{f}) \odot \mathcal{F}_T(\boldsymbol{c}).$$

4. So far, this is easy. But in order for $(\boldsymbol{z}, \boldsymbol{c}, \boldsymbol{m})$ to be a valid signature, it is necessary that $\boldsymbol{z}$ have fairly small coefficients. And finding a small'ish $\boldsymbol{z}$ with specified partial FFT is a hard problem.

Transcript Attacks

If each signature reveals a tiny, but nonzero, amount of information about the private key, then a DSS may be vulnerable to a transcript attack in which a long list of signatures is used to break the system.

The original PASSSign scheme had this weakness, as does NTRUSign. Various methods had been proposed that reduced (but never eliminated) the rate at which information leaks.

A few years ago Lyubashevsky described how to use **rejection sampling** to eliminate transcript leakage in certain lattice-based DSS.

We have adapted these ideas to make PASSSign immune to transcript attacks.

A Transcript Attack on Proto-PASSSign

Transcript attacks on proto-PASSSign use **vector reversals**

$$\overline{\boldsymbol{a}} = \overline{(a_0, \ldots, a_{N-1})} = (a_0, a_{N-1}, \ldots, a_1).$$

Let $(\boldsymbol{z}_1, \boldsymbol{c}_1, \boldsymbol{m}_1), (\boldsymbol{z}_2, \boldsymbol{c}_2, \boldsymbol{m}_2), \ldots$ be a list of signatures. Then one can show that

$$\text{Average of} \quad \boldsymbol{z}_i \star \overline{\boldsymbol{c}}_i \quad \text{converges to} \quad \kappa \boldsymbol{f} \star \overline{\boldsymbol{f}},$$

where $\kappa \neq 0$ is an easily computable constant. So a PASSSign transcript can be used to compute $\boldsymbol{f} \star \overline{\boldsymbol{f}}$, from which one can recover $\boldsymbol{f}$.

Rejection Sampling

The idea of rejection sampling is:

> When signing, if $\|\boldsymbol{z}\| > k - b$, reject the signature and sign again using a new nonce $\boldsymbol{r}$.

For appropriate choices of k and b, it may take 5 to 10 attempts to find a non-rejected signature.

With rejection sampling, one can show (under reasonable assumptions) that the signatures are *uniformly distributed* among all pairs of vectors

$$\{(\boldsymbol{z}, \boldsymbol{c}) : \|\boldsymbol{z}\| \leq k - b \text{ and } \boldsymbol{c} \in \mathbb{T}^N(b)\}.$$

Hence a transcript reveals *no information* about the private key $\boldsymbol{f}$.

Lattices Associated to Partial-FFTs

For simplicity, we will suppose that $T = \{0, 1, \ldots, t-1\}$. For a given public key $\mathcal{F}_T(\boldsymbol{f})$, consider the lattice $L_{\boldsymbol{f}}$ spanned by the rows of the $N+1+t$ dimensional matrix

$$\left[\begin{array}{ccccc|ccccc}
1 & 0 & \cdots & 0 & 0 & 1 & 1 & \cdots & & 1 \\
0 & 1 & \cdots & 0 & 0 & 1 & w & \cdots & & w^{t-1} \\
\vdots & & \ddots & \vdots & \vdots & \vdots & & \ddots & & \vdots \\
0 & 0 & \cdots & 1 & 0 & 1 & w^{N-1} & \ldots & & w^{(t-1)(N-1)} \\
\hline
0 & 0 & \cdots & 0 & 1 & \hat{f}_0 & \hat{f}_1 & & \cdots & \hat{f}_{t-1} \\
\hline
0 & 0 & \cdots & 0 & 0 & q & 0 & \cdots & & 0 \\
0 & 0 & \cdots & 0 & 0 & 0 & q & \cdots & & 0 \\
\vdots & & \ddots & \vdots & \vdots & \vdots & & \ddots & & \vdots \\
0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & & q
\end{array}\right]$$

The lattice $L_{\boldsymbol{f}}$ contains the short target vector

$$\tau_{\boldsymbol{f}} = [\boldsymbol{f} \mid 1 \mid \mathbf{0}].$$

Lattice Reduction Solution of SVR-PFFT

One can reduce the lattice dimension (without reducing the discriminant) by noting that $\tau_{\boldsymbol{f}}$ lies in

$$L'_{\boldsymbol{f}} = L \cap (\mathbb{R}^{N+1} \times \mathbf{0}).$$

Recovering $\tau_{\boldsymbol{f}}$ is a standard shortest vector problem whose difficulty may be estimated experimentally using (say) LLL-BKZ or other lattice reduction algorithms.

If we take $\#S = \#T \approx \frac{1}{2}N$, then recovering $\boldsymbol{f}$ and $\boldsymbol{m}$ are equally difficult, and our lattice problem has

$$\dim(L'_{\boldsymbol{f}}) \approx N \quad \text{and} \quad \operatorname{Disc}(L') \approx q^{N/2},$$

so

$$\frac{\text{length of target } \tau_{\boldsymbol{f}}}{\text{Gaussian expected value}} \approx \sqrt{\frac{\pi e}{2q}} \approx \frac{2}{\sqrt{q}}.$$

Directions and Questions

- The FFT is a ring homomorphism, and inversion of the partial-FFT of small vectors is a hard problem. So SVR-PFFT seems like a natural candidate for (leveled) homomorphic encryption.
- There is a hybrid PASS–NTRU encryption scheme that has some interesting operating characteristics in terms of tradeoffs between the difficulty of key recovery and plaintext recovery via lattice reduction.
- Both PASSEncrypt and PASSSign are "quantum resistant" in the sense that there are no known quantum algorithms to efficiently solve their underlying hard (lattice) problems.

I want to thank you for your attention, and Kristen Lauter and her group for inviting me to speak.

Public Key Cryptography Based on Partial Knowledge of Finite Fourier Transforms

Joseph H. Silverman

Brown University

Joint work with J. Hoffstein and J. Pipher (Brown University) and J. Schanck and W. Whyte (Security Innovations)

Colloquium, Microsoft Research, Redmond, WA

July 2013