RUHR-UNIVERSITÄT BOCHUM

Side-Channel Attacks On Implementations Of Lattice-Based Cryptosystems

Julian Speith, Felix Haarmann

Seminarausarbeitung
June 7, 2016
Embedded Security Group - Prof. Dr.-Ing. Christof Paar

Abstract

The following text[1] describes what your abstract should be about.

The abstract should convey to the reader concisely and accurately within the space of a few sentences, the claim to knowledge that the authors are making. It should indicate the boundaries of space and time within which the enquiry has occurred. If there is a claim to generality beyond the boundaries of the enquiry the basis of that claim should be given, for example that a random sample is thought to be representative of a larger population. There should also be a hint of the method of enquiry.

The boundaries of an enquiry are important - and are unfortunately too often omitted from abstracts. This is due to the regrettable tendency for researchers to generalise their results from, for example, a few schools to all schools, and to imply that what was true at a particular time, is true for all time. Some reference to the geographical location of the children, or teachers, or schools on whom the claim to knowledge rests should be made. Because of the international nature of the research community it is worth making clear in what country the research took place. Also the period in which the data was collected should be stated.

The abstract should be a condensation of the substance of the paper, not a trailer, nor an introduction. Journals and thesis regulations usually put a limit of around 200 to 300 words to the length of an abstract. Trailer is a term borrowed from the cinema industry to describe a showing of a few highlights in order to win an audience. An Introduction tells that something is coming, but doesnt reveal its substance. These are not what is needed.

Abstracts are recycled in abstract journals and electronic networks and provide the main vehicle for other researchers to become aware of particular studies. Hence the more clearly they convey the claim to knowledge of the original paper the more useful they are in helping the reader to decide whether it is worth taking the trouble to obtain and read the original and possibly cite it in his/her own writing.

Both the abstract and the paper should make sense without the other.

[1] Shamelessly ripped from `http://www.leeds.ac.uk/educol/abstract.htm`

Contents

Acronyms

1. Introduction

Despite the rapid progress in the development of quantum computers and the hereby increasingly urgent need for post-quantum cryptographic algorithms, no such algorithms has yet been standardized [CJL+16]. Current public-key cryptosystems like RSA, DHKE or even elliptic curve cryptography could easily be broken by a quantum computer, due to Shor's algorithm for prime factorization and discrete logarothms [Sho97]. As most of todays digital infrastructure depends (at least partially) on such public-key algorithms, the need for efficient and secure cryptography, that can withstand the power of quantum computation, is as high as never before.
Lattice-based cryptography is the most promising of all areas in post-quantum cryptography, as its underlying mathematics are already well understood and reasonably efficient implementations of some of the proposed cryptographic schemes are yet available. Our paper will give an overview over some selected lattice-based algorithms and their implementation in respect to their resistance to various side-channel attack techniques.

1.1. Related Work

Einfach kurz auf unsere 3-4 Paper eingehen, kommt noch... :D

1.2. Structure of this Paper

In Section 2 we will start with an explaination of our notation and give an overview over the mathematic background needed to understand this paper. This includes introducing the reader to the concept of *(ideal) lattices*, the *learning with errors problem (LWE)*, *Discrete Gaussian Distributions*, the *ring-LWE Encryption Scheme* and the *BLISS Signature Scheme*. Additionally, we will give a short explanation of the side-channel attack terminology used throughout this paper.
Section 3 will deal with the ring-LWE encryption scheme and will be split into two parts, starting with the description of a masked implementation of the decryption function, including a masked decoder build upon a masked table lookup. The

second part of this section will be an evaluation of the proposed implementation in respect to its soundness to first- and second-order side-channel attacks.
A different approach to masking of the ring-LWE encryption scheme will be presented in Section 4 of our paper, which will as well be split into a description of the proposed scheme and an evaluation. Furthermore, the second masking scheme will be compared to the first one in respect to efficiency and complexity.
Section 5 will discuss the `FLUSH+RELOAD` cache attacks on the Gaussian sampler used in the BLISS signature scheme. This part will start with a description of a perfect side channel attack on two Gaussian sampling algorithms, namely the cumulative distribution function (CDT sampling) and rejection sampling. This will be followed by an evaluation of the `FLUSH+RELOAD` attacks on an actual BLISS implementation, while running on modern CPUs.
Furthermore, in Section 6 we will be presenting two measures used for blinding polynomial multiplication and Gaussian sampling, which might help against the attacks described in Section 5.
Finally, Section 7 will summarize the content of our paper shortly and some conclusions will be drawn.

2. Theoretical Background

2.1. Notation

As we will only work with ideal lattices in our paper, all operations will be done within the ring $R_q = \mathbb{Z}_q[x]/(f(x))$ with $f(x)$ being an irreducible polynomial of degree n and all coefficients being reduced modulo q.
Polynomials will be written as $f(x)$ and vectors will be denoted by bold lower case letters, while matrices will be denoted by bold upper case letters. The entries of a vector $\mathbf{x}$ will be called x_i, with i specifying the position within the vector starting at 0.
The notation for the l_p norm of a vector $\mathbf{x}$ will be $\|x\|_p$, only with the exception of the l_2 norm, which will be referred to as $\|x\| = \sqrt{\sum_i x_i^2}$.

2.2. Ideal Lattices

A lattice Λ is discrete subgroup of $\mathbb{R}^n$ that is defined as a set of $m \leq n$ linearly independent vectors $\mathbf{b}_1, ..., \mathbf{b}_m \in \mathbb{R}^n$ and is generated by all linear combinations of those $\mathbf{b}_i$'s with integer coefficients:

$$\Lambda(\mathbf{b}_1, ..., \mathbf{b}_m) = \left\{ \sum_{i=1}^{m} x_i \mathbf{b}_i \;\middle|\; x_i \in \mathbb{Z} \right\}$$

The set $\{\mathbf{b}_1, ..., \mathbf{b}_m\}$ of those vectors is called the basis of that lattice. Such a basis is commonly represented by a matrix $\mathbf{B} = (\mathbf{b}_1, ..., \mathbf{b}_m)$.
Furthermore, an ideal lattice is is lattice that corresponds to ideals in a ring R_q. This basically means, that we can deal with polynomials instead of matrices, which makes arithmetics used for cryptographic applications much more efficient. In our paper we will confine ourselves to those ideal lattices, as most of the current work in that area focuses around them. For more on the topic of ideal lattices, see [LPR12].

2.3. Learning with Errors Problem

2.4. Discrete Gaussian Distribution

The discrete Gaussian distribution with mean μ and standard deviation σ is denoted as $\mathcal{N}_{\mathbb{Z}}(\mu, \sigma^2)$. In this paper we will focus on zero-centered distributions $\mathcal{N}_{\mathbb{Z}}(0, \sigma^2)$ with a density function $\rho_\sigma(x)$ given by:

$$\rho_\sigma(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}$$

The probability function of discrete Gaussian distribution over $\mathbb{Z}$ is then defined as $D_\sigma(x) = \rho_\sigma(x)/\rho_\sigma(\mathbb{Z})$ with $\rho_\sigma(\mathbb{Z}) = \sum_{y=-\infty}^{\infty} \rho_\sigma(y)$. As we will sample whole vectors most of the time, we denote the discrete Gaussian distribution over $\mathbb{Z}^m$ as $\mathcal{N}_{\mathbb{Z}}^n(\mu, \sigma^2)$ and its probability function as $D_\sigma^m(\mathbf{x}) = \rho_\sigma(\mathbf{x})/\rho_\sigma(\mathbb{Z})^m$ with $\rho_\sigma(\mathbf{x})$ being defined as follows:

$$\rho_\sigma(\mathbf{x}) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\|\mathbf{x}\|^2}{2\sigma^2}}$$

2.5. Cryptographic Algorithms

2.5.1. Ring-LWE Encryption Scheme

2.5.2. BLISS Signature Scheme

Algorithm 1 BLISS Key Generation

Output: BLISS key pair $(\mathbf{A}, \mathbf{S})$ with public key $\mathbf{A} = (\mathbf{a}_1, \mathbf{a}_2) \in R_{2q}^2$ and secret key $\mathbf{S} = (\mathbf{s}_1, \mathbf{s}_2) \in R_{2q}^2$, such that $\mathbf{AS} = \mathbf{a}_1 \cdot \mathbf{s}_1 + \mathbf{a}_2 \cdot \mathbf{s}_2 \equiv q \bmod 2q$

1: Choose $\mathbf{f}, \mathbf{g} \in R_{2q}$ uniformly at random with exactly d_1 entries in $\{\pm 1\}$ and d_1 entries in $\{\pm 2\}$
2: $\mathbf{S} = (\mathbf{s}_1, \mathbf{s}_2) = (\mathbf{f}, 2\mathbf{g} + 1)$
3: **if** $\mathbf{f}$ violates certain conditions (see [DDLL13]) **then**
4: Restart
5: **end if**
6: $\mathbf{a}_q = (2\mathbf{g} + 1)/\mathbf{f} \bmod q$ (restart if $\mathbf{f}$ is not invertible)
7: **return** $(\mathbf{A}, \mathbf{S})$ with $\mathbf{A} = (2\mathbf{a}_q, q - 2) \bmod 2q$

2.6. Side-Channel Attack Terminology

Algorithm 2 BLISS Signature Algorithm

Input: Message μ, public key $\mathbf{A} = (\mathbf{a}_1, q-2)$, secret key $\mathbf{S} = (\mathbf{s}_1, \mathbf{s}_2)$
Output: Signature $(\mathbf{z}_1, \mathbf{z}_2^\dagger, \mathbf{c}) \in \mathbb{Z}_{2q}^n \times \mathbb{Z}_p^n \times \{0,1\}^n$

1: $\mathbf{y}_1, \mathbf{y}_2 \leftarrow D_\sigma^n$
2: $\mathbf{u} = \zeta \cdot \mathbf{a}_1 \cdot \mathbf{y}_1 + \mathbf{y}_2 \bmod 2q$
3: $\mathbf{c} = H(\lfloor \mathbf{u} \rceil_d) \bmod p, \mu$
4: Choose a random bit b
5: $\mathbf{z}_1 = \mathbf{y}_1 + (-1)^b \mathbf{s}_1 \cdot \mathbf{c} \bmod 2q$
6: $\mathbf{z}_2 = \mathbf{y}_2 + (-1)^b \mathbf{s}_2 \cdot \mathbf{c} \bmod 2q$
7: Continue with a probability based on σ, $\|\mathbf{Sc}\|$, $\langle \mathbf{z}, \mathbf{Sc} \rangle$ (see [DDLL13]), else restart
8: $\mathbf{z}_2^\dagger = (\lfloor \mathbf{u} \rceil_d - \lfloor \mathbf{u} - \mathbf{z}_2 \rceil_d) \bmod p$
9: **return** $(\mathbf{z}_1, \mathbf{z}_2^\dagger, \mathbf{c})$

Algorithm 3 BLISS Verification Algorithm

Input: Message μ, public key $\mathbf{A} = (\mathbf{a}_1, q-2)$, signature $(\mathbf{z}_1, \mathbf{z}_2^\dagger, \mathbf{c})$

1: **if** $\mathbf{z}_1, \mathbf{z}_2^\dagger$ violate certain conditions (see [DDLL13]) **then**
2: Reject
3: **end if**
4: **if** $\mathbf{c} = H(\lfloor \zeta \cdot \mathbf{a}_1 \cdot \mathbf{z}_1 + \zeta \cdot q \cdot \mathbf{c} \rceil_d + \mathbf{z}_2^\dagger \bmod p, \mu)$ **then**
5: Accept
6: **end if**

3. Masking the Ring-LWE Encryption Scheme I

3.1. Implementation

3.2. Evaluation

4. Masking the Ring-LWE Encryption Scheme II

4.1. Implementation

4.2. Evaluation

5. Flush+Reload Cache Attack on Bliss

5.1. Gaussian Sampling

5.1.1. CDT Sampling

5.1.2. Rejection Sampling

5.2. Attacking the Sampling Algorithms

5.3. Evaluation

6. Blinding Countermeasures

Blinding is a countermeasure commonly used to prevent side-channel attacks like *Differential Power Analysis* (DPA) [KJJ99]. It is used to add additional randomness to mathematical operations in a way, that the attacker can not easily draw conclusions from his observations. This Section summarizes two blinding countermeasures presented in [Saa16], which appear to be of special interest for Ring-LWE cryptosystems. While the first countermeasure will be an approach to blinding of polynomial multiplication within a ring R_q, the second one will be a blinding countermeasure for Gaussian sampling.

6.1. Blinding Polynomial Multiplication

There are two pretty types of blinding for polynomial multiplications, the first of whom is the multiplication of each polinomial with a constant. For two polynomials $f, g \in R_q$ and constants $a, b \in \mathbb{Z}_q$ the blinding operation and the inverse operation look as follows:

$$h(x) = af(x) \cdot bg(x)$$
$$f(x) \cdot g(x) = (ab)^{-1}h(x)$$

The second type would be circularly shifting the coefficients in each of the polynomials. As a polinomial can be written as $f(x) = \sum_{i=0}^{n-1} f_i x^i$, a shift by j positions would be equal to the following computation:

$$x^j f(x) = \sum_{i=0}^{n-1} f_i x^{i+j} = \sum_{i=0}^{n-1} f_{i-j} x^i$$

Both of those blinding operation can be combined within one function, which will be called PolyBlind$(\mathbf{v}, s, c)$ from now on. This function works on coefficient vectors of length n instead of the polynomials themselves and is given in Algorithm 4.

The inverse operation can be denoted by PolyBlind$(\mathbf{v'}, -s, c^{-1})$. Due to the isometries of the ring R_q, the multiplication of two polynomials (here: their coefficient vectors) can be blinded using the PolyBlind function in the following way:

Algorithm 4 PolyBlind

Input: coefficient vector **v**, number of shifts s, constant c
Output: blinded coefficient vector **v'**

1: **for** $i = 0, ..., n - s - 1$ **do**
2: $\quad v'_i = cv_{i+s} \bmod q$
3: **end for**
4: **for** $i = n - s, ..., n - 1$ **do**
5: $\quad v'_i = q - cv_{i+s-n} \bmod q$
6: **end for**
7: **return v'**

$$\mathbf{f'} = \text{PolyBlind}(\mathbf{f}, r, a) \text{ with } r \in_R 0, ..., n-1 \text{ and } a \in_R \mathbb{Z}_q$$
$$\mathbf{g'} = \text{PolyBlind}(\mathbf{g}, s, b) \text{ with } s \in_R 0, ..., n-1 \text{ and } b \in_R \mathbb{Z}_q$$
$$\mathbf{h'} = \mathbf{f'} \cdot \mathbf{g'}$$
$$\mathbf{h} = \text{PolyBlind}(\mathbf{h'}, -(r+s), (ab)^{-1})$$

6.2. Blinding Gaussian Sampling

As with the blinding of polynomial multiplication in the last subsection, there are two pretty easy ways to blind the coefficient vectors during the process of Gaussian sampling. We will again give a short description of both of them and present a function, that combines both methods.
We define a function VectorSample(n, σ), that samples and returns a vector according to the discrete Gaussian distribution $\mathcal{N}^n_{\mathbb{Z}}(0, \sigma^2)$. A naiv implementation of this function could lead to leakage of information to an attacker using e.g. DPA. This has been done in the cache attack from [BHLY16] we described in Section 5.
The first approach to blinding would be to randomly shuffle the elements in the coeffcient vector. The function VectorShuffle$(\mathbf{x})$ is doing exactly that, so that VectorShuffle(VectorSample(n, σ)) would increase security to a certain extend.
For the second approach we need to take a short detour through probability theory. For two Gaussian distributions $X = \mathcal{N}^n_{\mathbb{Z}}(\mu_X, \sigma^2_X)$ and $Y = \mathcal{N}^n_{\mathbb{Z}}(\mu_Y, \sigma^2_Y)$, we know that their sum is equal to $X + Y = \mathcal{N}^n_{\mathbb{Z}}(\mu_X + \mu_Y, \sigma^2_X + \sigma^2_Y)$. As we focus on zero-centered distributions, the center does not change for us. For the standard deviation it follows, that $\sigma_{X+Y} = \sqrt{\sigma^2_x + \sigma^2_Y}$.

Algorithm 5 VectorBlindSample

Input: length of the vector n, number of iterations m, standard deviation σ
Output: sampled vector $\mathbf{x}$

1: $\mathbf{x} = \mathbf{0}$
2: **for** $i = 1, ..., m$ **do**
3: $\quad \mathbf{x} = \mathbf{x} + \mathcal{N}_{\mathbb{Z}}^{n}(0, (\frac{1}{\sqrt{m}}\sigma)^2)$
4: $\quad \mathbf{x} = \text{VectorShuffle}(\mathbf{x})$
5: **end for**
6: **return x**

7. Conclusion

Conclude your thesis and discuss your results...

List of Figures

List of Tables

A. Bibliography

[BHLY16] Leon Groot Bruinderink, Andreas Hlsing, Tanja Lange, and Yuval Yarom. Flush, gauss, and reload – a cache attack on the bliss lattice-based signature scheme. Cryptology ePrint Archive, Report 2016/300, 2016.

[CJL+16] Lily Chen, Stephen Jordan, Yi-Kai Liu, Dustin Moody, Rene Peralta, Ray Perlner, and Daniel Smith-Tone. Report on post-quantum cryptography. Technical Report NIST IR 8105, National Institute of Standards and Technology (NIST), February 2016.

[DDLL13] Lo Ducas, Alain Durmus, Tancrde Lepoint, and Vadim Lyubashevsky. Lattice signatures and bimodal gaussians. Cryptology ePrint Archive, Report 2013/383, 2013.

[KJJ99] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, pages 388–397, 1999.

[LPR12] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. Cryptology ePrint Archive, Report 2012/230, 2012.

[Saa16] Markku-Juhani O. Saarinen. Arithmetic coding and blinding countermeasures for ring-lwe. Cryptology ePrint Archive, Report 2016/276, 2016.

[Sho97] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509, October 1997.