Partial Key Exposure Attacks on Multi-power RSA

A thesis submitted to the
Graduate School of Natural and Applied Sciences

by

Muhammed Fethullah ESGİN

in partial fulfillment for the
degree of Master of Science

in
Cybersecurity Engineering

İSTANBUL
ŞEHİR
ÜNIVERSITY

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science in Cybersecurity Engineering.

APPROVED BY:

Dr. Osmanbey Uzunkol ....
(Thesis Advisor)

Dr. Mehmet Sabır Kiraz ....
(Thesis Co-advisor)

Prof. Dr. Erkan Türe ....

Asst. Prof. Dr. Seher Tutdere ....

Dr. Kerem Kaşkaloğlu ....

This is to confirm that this thesis complies with all the standards set by the Graduate School of Natural and Applied Sciences of İstanbul Şehir University:

DATE OF APPROVAL: 16/06/2015

SEAL/SIGNATURE:

Declaration of Authorship

I, Muhammed Fethullah ESGİN, declare that this thesis titled, 'Partial Key Exposure Attacks on Multi-power RSA' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: M.

Date: 16 06 2015

Partial Key Exposure Attacks on Multi-power RSA

Muhammed Fethullah ESGİN

Abstract

In this thesis, our main focus is a type of cryptanalysis of a variant of RSA, namely multi-power RSA. In multi-power RSA, the modulus is chosen as $N = p^r q$, where $r \geq 2$. Building on Coppersmith's method of finding small roots of polynomials, Boneh and Durfee show a very crucial result (a small private exponent attack) for standard RSA. According to this study, $N = pq$ can be factored in polynomial time in $\log N$ when $d < N^{0.292}$. In 2014, Sarkar improve the existing small private exponent attacks on multi-power RSA for $r \leq 5$. He shows that one can factor N in polynomial time in $\log N$ if $d < N^{0.395}$ for $r = 2$.

Extending the ideas in Sarkar's work, we develop a new partial key exposure attack on multi-power RSA. Prior knowledge of least significant bits (LSBs) of the private exponent d is required to realize this attack. Our result is a generalization of Sarkar's result, and his result can be seen as a corollary of our result. Our attack has the following properties: the required known part of LSBs becomes smaller in the size of the public exponent e and it works for all exponents e (resp. d) when the exponent d (resp. e) has full-size bit length. For practical validation of our attack, we demonstrate several computer algebra experiments. In the experiments, we use the LLL algorithm and Gröbner basis computation. We achieve to obtain better experimental results than our theoretical result indicates for some cases.

Keywords: Cryptography, Public-key Cryptography, RSA, Multi-power RSA, Partial Key Exposure, Lattice Reduction, LLL, Coppersmith's Method

Çoklu Kuvvet RSA'ya Kısmi Bilgi Saldırıları

Muhammed Fethullah ESGİN

Öz

Bu tezde temel olarak bir RSA çeşidinin kriptoanalizi üzerine yoğunlaşıyoruz. Çoklu kuvvet RSA olarak adlandırılan bu çeşitte RSA modülü $N = p^r q$, $r \geq 2$, olacak şekilde seçilmektedir. Boneh ve Durfee standart RSA üzerinde Coppersmith'in polinomların küçük köklerini bulma yöntemini kullanarak çok önemli bir sonuç (bir küçük gizli üs saldırısı) göstermişlerdir. Bu çalışmaya göre $N = pq$ için $d < N^{0.292}$ sağlandığında $\log N$ üzerinden polinom zamanda N sayısı çarpanlarına ayrılabilmektedir. 2014'te, Sarkar çoklu kuvvet RSA üzerinde var olan küçük gizli üs saldırılarını $r \leq 5$ için geliştirmiştir. Burada $r = 2$ için $d < N^{0.395}$ sağlanması durumunda N'nin $\log N$ üzerinden polinom zamanda çarpanlarının bulunabileceği gösterilmiştir.

Biz, Sarkar'ın çalışmasındaki fikirleri genişleterek çoklu kuvvet RSA'ya yeni bir kısmi bilgi saldırısı geliştireceğiz. Saldırının gerçekleştirilebilmesi için gizli üs d'nin en önemsiz bitlerinin bir kısmının bilinmesi gerekmektedir. Bulduğumuz sonuç, Sarkar'ın sonucunun bir genelleştirmesi olup, Sarkar'ın sonucu bizim sonucumuzun bir yan sonucu olarak görülebilmektedir. Saldırımız şu özellikleri taşımaktadır: gerekli olan en önemsiz bit miktarı açık üs e küçüldükçe azalmaktadır ve d'nin (e'nin) bit boyu N ile aynı olsa dahi bütün üsler e (d) için çalışmaktadır. Saldırımızın pratik olarak çalıştığını göstermek için bilgisayar üzerinde bazı cebirsel deneyler gösterilmiştir. Deneylerde LLL algoritması ve Gröbner bazı hesaplaması kullanılmaktadır. Bazı deneylerde teorik sonucumuzun belirttiğinden daha iyi deney sonuçları elde edilmiştir.

Anahtar Sözcükler: Kriptografi, Açık Anahtarlı Kriptografi, RSA, Çoklu Kuvvet RSA, Kısmi Bilgi Saldıları, Örgü İndirgeme, LLL, Coppersmith'in Yöntemi

Acknowledgments

First of all, I would like to thank my advisors, Osmanbey Uzunkol and Mehmet Sabır Kiraz, for their support throughout this thesis. They initiated my research about the public-key cryptography. They would always send me interesting new results about RSA and related subjects. This led me to choose this topic for my thesis. My thesis topic would have probably been a different one without their help. I'm also thankful to them for our long discussions about various research topics, and for their help in improving my writing skills especially for an introduction part of a manuscript. Thank you for your constructive criticisms.

I would also like to thank all my colleagues at TÜBİTAK BİLGEM for their help and for making me want to come to work everyday with great pleasure. I thank our managers, Fatih Birinci and Orhun Kara, for making our unit such an enjoyable environment.

Finally, I want to thank the committee members of my thesis, Erkan Türe, Seher Tutdere and Kerem Kaşkaloğlu, for their help in improving this thesis.

Contents

List of Figures

List of Tables

Abbreviations

IFP	**I**nteger **F**actorization **P**roblem
DLP	**D**iscrete **L**ogarithm **P**roblem
ECDLP	**E**lliptic **C**urve **D**iscrete **L**ogarithm **P**roblem
CRT	**C**hinese **R**emainder **T**heorem
GCD	**G**reatest **C**ommon **D**ivisor
NFS	**N**umber **F**ield **S**ieve
ECM	**E**lliptic **C**urve **M**ethod
PKCS	**P**ublic **K**ey **C**ryptography **S**tandard
MSB	**M**ost **S**ignificant **B**it
LSB	**L**east **S**ignificant **B**it

Chapter 1

Introduction

As we are in the era of technology, the power of knowledge is of the utmost importance. Everyone from high-valued companies to the Internet users require to secure their private data, and the key to *information security* is *cryptology.* Information security is not only a reference to enciphering data but also has other aspects such as *integrity*, *authentication* and *non-repudiation.* These properties are all realized via cryptographic solutions. Every online banking, instant messaging or e-trade user, with or without knowing it, benefits from various cryptographic solutions. First of all, an online banking user must be ensured that no unauthorized party learns his password (*confidentiality*). Then, he/she must be certain that every action he/she does must be received by the bank as intended (integrity). Moreover, the bank must know that if someone makes an order, he/she cannot later deny ever doing it (non-repudiation). All these concerns form the basis for cryptology.

Cryptology is defined as the study of hiding information in the presence of an untrusted party, and consists mainly of two parts: *cryptography* and *cryptanalysis.* Main concern of cryptography is designing new cryptosystems that enable the users to hide some sensitive information. Such a property of hiding private information is referred as *confidentiality* and this can be achieved by enciphering (or encrypting) the information at hand. As mentioned, there are more *security services* concerning cryptography such as integrity, authentication and non-repudiation. These services are provided using cryptographic tools such as digital signatures and hash algorithms (see [1] for details).

In cryptanalysis, on the other hand, one tries to break a cryptographic system by any means. It can be the prevention of any one of the security services. So, the objective of an attacker can, for example, be one of the followings: decrypting encrypted data, modifying encrypted data in a way that it is validated in a check process, posing as someone else, tracking users and revealing the identity of an anonymous entity. In fact, we can say that cryptography contains cryptanalysis because it is impossible to design a new system without making a thorough analysis of it.

In general, a cryptographic scheme is illustrated as follows: There are 2 parties, Alice and Bob. They try to communicate via a (potentially) insecure channel in the presence of an adversary, Eve, who tries to acquire some knowledge about the communication between Alice and Bob. Till the invention of *public-key cryptography* in 1976 [2], it was always assumed that Alice and Bob shared a *secret key*, which is unknown to the adversary, Eve. In that way, using this shared secret key (or also referred as *private key*), Alice can encrypt messages that she wants to send and decrypt messages from Bob, and vice versa. Such a cipher having only one secret key (or keys that can easily be derived knowing only one of them) is called a *symmetric-key cipher* (see [1] for a more detailed overview of cryptology).

In 1976, Whitfield Diffie and Martin Hellman published their paper “New Directions in Cryptography” [2] that opened up, as its title suggests, new directions in cryptography. This paper is the first publicly known introduction of asymmetric (or public-key) cryptography. They describe a method so that two parties can agree on a secret key over an insecure channel. In public-key cryptography, Alice possesses a key pair consisting of a *public key* and a *private key*. She publishes her public key so that everyone including Bob and Eve can send her encrypted messages using this public key. However, as Alice keeps the private key to herself, only she can decrypt messages that are encrypted under her public key. The system in this case relies on the hardness (computational infeasibility) of finding the private key when the public key is known. This is usually referred as a *trapdoor* mechanism. That is decryption is hard knowing the public key but easy knowing the private key.

There are mainly two mathematical problems upon which public-key scheme are built: the Integer Factorization Problem (IFP) and the Discrete Logarithm Problem (DLP). In the IFP, one is given an integer $N = pq$ that is the product of two (distinct) primes p

and q. The aim here is then to find prime factors p (or q). In the DLP, there is a cyclic group $G =< g >$ and an element $h \in G$. This implies that (using the multiplicative notation) there exists an $x \in \mathbb{Z}$ such that $g^x = h$. The problem is then to find $x = log_g h$ given G, g and h.

The most famous IFP-based asymmetric crypto scheme is the RSA cryptosystem [3]. Famous DLP-based cryptographic schemes are Diffie-Hellman key exchange protocol [2], ElGamal encryption scheme [4] and the Elliptic Curve Discrete Logarithm Problem (ECDLP) based algorithms [5, 6]. The main disadvantage of the IFP over the DLP is that, in the IFP, one necessarily works on a ring that may enable an attacker to exploit structural properties of the ring. In the DLP, however, a group structure is enough, which seriously limits the capabilities of the attacker. This is the main reason behind the belief that much shorter key size for the ECDLP based algorithms provide same security level with longer key sizes for RSA. For example, 512-bit key for an elliptic curve is assumed to be equivalent to 15360-bit RSA key [7]. In Table 1.1, different key sizes of symmetric-key ciphers, the ECDLP-based algorithms and the IFP-based algorithms are compared [7].

Table 1.1: Comparison of different key sizes between symmetric-key ciphers, the ECDLP-based algorithms and the IFP-based algorithms.

	Symmetric-key	ECDLP-based	IFP-based
	80	160	1024
Key	128	256	3072
Size	192	384	7680
	256	512	15360

In most of the cryptographic systems, a hybrid system is adopted. A public-key scheme is used to agree on a symmetric key, and then symmetric-key systems with that key are used in the next stages. The reason for this is that in symmetric-key cryptography parties sharing encrypted information need to agree on a private key, which should not be acquired by anyone else. That means either sharing the key via a secure channel or using public-key cryptography. Additionally, the reason for continuing the secure communication using symmetric-key ciphers is that they work much faster compared to public-key algorithms. This is due to fact that the mathematical problems that asymmetric algorithms rely on are usually solvable in subexponential time in the bit length of the key (ECDLP being an exception) while the security margin grows exponentially in the bit length of the key in symmetric algorithms. Hence, having a larger key results in higher computational cost.

In this work, we will be mainly interested in public-key cryptography, specifically the RSA cryptosystem. RSA is the first public-key encryption algorithm developed in 1978 by Rivest, Shamir and Adleman [3], and the most commonly used one ever since its introduction. We can state two main reasons why it is more commonly used than the other public-key schemes: easiness and simple implementation. Because, it does not require deep mathematical background to implement it (as we will explain in Section 2.1). Due to being in the core of public-key cryptography, it has attracted a lot of attention of various cryptanalysts from all over the world. Boneh's work [8] in 1999 about previous cryptanalysis of RSA may be a good guideline for understanding the cryptanalytic path of the RSA cryptosystem. For a more recent and detailed survey on the cryptanalysis of RSA, we refer to [9].

In 1990, Wiener described an attack on RSA [10] that shows RSA is easily broken whenever the secret key d satisfies $d < \frac{1}{3}N^{\frac{1}{4}}$ where N represents the RSA modulus. This attack shows that under certain conditions RSA may be very weak. This opened up a very important question what other possible conditions are (if any) such that RSA is weak. In this thesis, we also concentrate on this question. However, the attacks will require not only *implicit knowledge* (as in the case of Wiener's attack) but also *explicit knowledge*. What we mean by an implicit knowledge is that the adversary knows that secret information (keys) satisfies certain conditions. On the other hand, knowing the value of some part of the secret information is considered as explicit knowledge.

1.1 A Short History of the Partial Key Exposure Attacks

Standard RSA parameters can be described as follows. Let $N = pq$ be the product of two different large primes. Choose two integers e and d such that d is the inverse of e modulo $\phi(N) = (p-1)(q-1)$. Then, (N, e) is the public key and d (or (d, p, q)) is the private key. Now, the encryption and decryption work as follows:

$$E(m) := m^e \mod N,$$

$$D(c) := c^d \mod N.$$

Here, N is referred as the RSA modulus, e the public exponent and d the private exponent.

Partial key exposure attacks were first suggested by Boneh, Durfee and Frankel [11] in 1998. The attack can be described as finding the whole secret exponent d in RSA with the help of a partial knowledge of the secret exponent. The main tool used in the attack is Coppersmith's algorithms for finding small roots of polynomials [12–14], which takes advantage of lattice reduction techniques (in particular, the LLL algorithm [15]). The benefit of the attack is that it apriori tells the attacker if the attack is going to be successful. So, the attack actually predetermines how many bits of d the attacker has to acquire in order to successfully break the system. When first proposed by Boneh, Durfee and Frankel, the attack was only applicable when the public exponent e satisfies $e < \sqrt{N}$. However, the attack worked for $N^{\frac{1}{4}} < e < N^{\frac{1}{2}}$ only when the factorization of e is known in addition to the partial knowledge from the private key. They left it as an open question whether the attack could still be applied under weaker assumptions. Later in 2003, Blömer and May [16] increased the bound on e such that their strongest attack worked when $e < N^{\frac{7}{8}}$.

This led to the belief that the attack should work for all e up to full-size (i.e., the bitsize of e can be as large as the bitsize of N). Not long after, it turned out that the attack in fact worked for all e up to full-size [17]. However, as we employ the attack with weaker assumptions, it becomes that we need more bits of d known to reconstruct all of it. After, showing that a partial key exposure attack can be mounted for any e, the works started to concentrate more on employing the attack on different variants of RSA. Especially, CRT-RSA [18] is studied widely. There is also an interesting result by Joye and Lepoint [19] that shows a partial key exposure attack can be mounted on RSA with private exponents larger than N.

Remark 1.1. There is a technique called *private exponent blinding* that is used to protect against *side-channel attacks* (see Section 2.4.2.1). In this technique, $d' = d + a\phi(N)$, for some $a \in \mathbb{Z}$, is used as the private exponent to disable an attacker from learning the bits of d by observing the bits of d'.

Attacks using Coppersmith's methods are highly dependent on the polynomial f whose roots are to be found. Today, there is no provable method providing the optimal conditions for such attacks. Due to this dependence on f cryptanalysts to find a general strategy are limited which results mostly in exploring ad hoc methods for lattice constructions. In [20, 21] a general strategy for a lattice construction is explained. However, this does not always provide the optimal bounds, and is not always compatible with different techniques.

1.2 Overview of the Thesis

Research goal:

Our goal in this thesis is to explore new partial key exposure attacks on multi-power RSA. The main assumption is that an attacker acquires some least significant bits of the private exponent d. The attack aims at factoring the modulus $N = p^r q$, where $r \geq 2$.

Organization of the thesis:

Chapter 2: The RSA Cryptosystem

To begin with, we give an overview of the RSA cryptosystem and one of its many variants, multi-power RSA. An important part of this chapter consists of various cryptanalysis of RSA. The attacks include those that form the basis for partial key exposure attacks: Wiener's attack [10] and Boneh-Durfee attack [22]. Actually, Boneh-Durfee attack is studied in detail in Chapter 3.

Chapter 3: Preliminaries

In Chapter 3, we first outline some preliminaries about the lattice theory. Then, we concentrate on Coppersmith's method of finding small roots of polynomials [12–14]. This is core of partial key exposure attacks. Since we use the method of finding small modular roots, this one is studied in more detail. Later, we discuss the complexity of attacks using Coppersmith's methods. The chapter is concluded with the introduction of Boneh-Durfee attack [22].

Chapter 4: Partial Key Exposure Attacks on Multi-power RSA

In Chapter 4, we first discuss previous studies using Coppersmith's methods of finding small roots of polynomials on multi-power RSA. These include small private exponent attacks [23–27] and partial key exposure attacks [27, 28]. Later, the main contribution of this thesis is given. That is, we show a new partial key exposure attack on multi-power RSA. The attack works even when the exponents satisfy $e, d \approx N$.

A part of this chapter is based on [29], which has been accepted to CAI 2015 (joint work with the supervisor and the co-supervisor).

Chapter 5: Conclusion and Discussions

Chapter 5 is the final chapter of this thesis. First, a short overview this work is given. Then, we argue some issues about a partial key exposure attack in [28]. Finally, we conclude the thesis with some discussions and further study options/open problems in the area of the thesis.

Chapter 2

The RSA Cryptosystem

In this chapter, we give details of the RSA cryptosystem [3], show how the system works, and explain also its multi-power RSA variant, about which we show a new partial key exposure attack in Chapter 4. Firstly, we start with the mathematical preliminaries of RSA parameters. Afterwords, we explain a method with which the key generation can be realized. We fix $\mathbb{Z}_N := \mathbb{Z}/N\mathbb{Z} = \{0, 1, \ldots, N-1\}$.

2.1 RSA

Let $N = pq$ be the product of two large distinct primes p and q. Further, let $1 < e < \phi(N)$ with $\gcd(e, \phi(N)) = 1$ where $\phi(N) := \#\{1 \leq k < N \mid (k, N) = 1\}$ and $\gcd(x, y)$ denotes the greatest common divisor of x and y. (ϕ is called the Euler's Totient function). Note that for such an N, $\phi(N) = (p-1)(q-1)$ by definition. Also, let d be the multiplicative inverse of e modulo $\phi(N)$. i.e., $ed \equiv 1 \bmod \phi(N)$. Now, the encryption of a plaintext $m \in \mathbb{Z}_N$, $\gcd(m, N) = 1$, and the decryption of a ciphertext $c \in \mathbb{Z}_N$, $\gcd(c, N) = 1$, are given as the following functions:

$$E(m) := m^e \mod N,$$
$$D(c) := c^d \mod N,$$

respectively. The relation between e and d yields to the following equation which we call the RSA equation:

$$\begin{aligned} & e^{-1} && \equiv d \mod \phi(N) \\ \Longrightarrow \quad & ed && \equiv 1 \mod \phi(N) \\ \Longrightarrow \quad & ed - 1 && = k.\phi(N) \text{ for some } k \in \mathbb{Z} \end{aligned} \tag{2.1}$$

In most of the partial key exposure attacks, equation (2.1) (or a variant of it) is used.

Theorem 2.1. *Let* $m \in \mathbb{Z}/N\mathbb{Z}$ *and* $c \equiv m^e \mod N$. *Then,* $c^d \equiv m \mod N$.

Proof. If $m = 0$, the result is clear. If $\gcd(m, N) = 1$, then $m \in (\mathbb{Z}/N\mathbb{Z})^*$ and

$$\begin{aligned} c^d & \equiv m^{ed} && \mod N \\ & \equiv m^{k.\phi(N)+1} && \mod N \\ & \equiv (m^{\phi(N)})^k m && \mod N \\ & \equiv m && \mod N \end{aligned}$$

since $|(\mathbb{Z}/N\mathbb{Z})^*| = \phi(N)$. If $\gcd(m, N) \neq 1$, then we can say that $m = ap$ for some $a > 0$ without loss of generality. Then, we get

$$\begin{aligned} c^d & \equiv m^{ed} && \mod q \\ & \equiv m^{k.(q-1)(p-1)+1} && \mod q \\ & \equiv m && \mod q && \text{(by Fermat's Little Theorem)} \end{aligned}$$

and also

$$c^d \equiv m^{ed} \equiv (ap)^{ed} \equiv 0 \mod p$$

Finally, using the Chinese Remainder Theorem, we see that $c^d \equiv m \mod N$. □

Note that although the above theorem is true for any $m \in \mathbb{Z}/N\mathbb{Z}$, $\gcd(m, N) \neq 1$ should not hold for a plaintext in RSA. Because, otherwise, an attacker can compute a prime factor of N by computing $\gcd(m, N)$.

2.2 RSA Key Generation

Here, we give a possible way of RSA key generation. We note that depending on the designer's criteria for parameters, different algorithms (probability with condition checks on d and/or e) have to be adopted.

1. Pick two large random primes p, q of the same size.
2. Set $N := pq$.
3. Choose an integer e such that $\gcd(e, \phi(N)) = 1$.
4. Find $d := e^{-1}$ modulo $\phi(N)$.
5. (e, N) is the public key and (d, p, q) (or just d) is the private key.

One may raise a question "can we easily find random primes of the same size?". The answer relies on the prime number theorem which states that for large $X \in \mathbb{R}$, there are approximately $\frac{X}{\ln X}$ prime numbers up to X [30]. Furthermore, finding an e with $\gcd(e, \phi(N)) = 1$ is computationally easy because we can just choose any prime number less than $\phi(N)$.

The bitsize n of N is usually referred as the *security parameter* because it determines the level of security of the underlying RSA system. Today, $n = 2048$ is believed to be secure.

2.3 Multi-power RSA (Takagi's Variant)

Multi-power RSA (also referred as Takagi's RSA or prime power RSA) is introduced by Takagi in [24]. One of the motivations of this variant is to speed up the RSA decryption/signing process. More concretely, $N = p^r q$ is chosen for two (distinct) primes of same bit length such that $r \geq 2$. Then, there are two different ways of generating public/private exponents. The first one imposes the condition $ed \equiv 1 \bmod (p-1)(q-1)$ while the other $ed \equiv 1 \bmod \phi(N)$, where $\phi(N) = p^{r-1}(p-1)(q-1)$. Decryption of a ciphertext c is computed more efficiently using simply a combination of Hensel lifting and Chinese Remainder Theorem modulo p^r and q (see [24] for details).

2.4 Cryptanalysis of RSA

In this section, we revisit some cryptanalytic methods on RSA. If two distinct RSA moduli N_1 and N_2 share a common prime factor, then this factor can be extracted easily by a greatest common divisor (gcd) computation between N_1 and N_2. However, for example, for two 1024-bit RSA moduli with random 512-bit prime factors, the probability that they share a common prime factor is negligibly low. That's why one suspects that if prime factors of two different RSA moduli coincide, there must be something wrong with the prime number generation. This is exactly the case in [31], [32] and [33]. In these papers, authors find common prime factors using the batch gcd algorithm (which is used to efficiently calculate gcd of many numbers) and show deficiencies in the prime number generation in these specific cases.

Another observation is that knowing $\phi(N)$ is equivalent to knowing p and q. This is due to the fact that $\phi(N) = (p-1)(q-1) = N - (p+q-1)$. Since N is already known, one can easily find $s := p + q$. After that all one needs to do is to find the roots of the polynomial $f(x) = x^2 - sx + N$, which yields p and q. Thus, $\phi(N)$ must be guarded as safe as the prime factors.

Lastly, it is easy to see that RSA has multiplicative homomorphic property. That is, for any x_1 and x_2,

$$\begin{aligned} & D(x_1) \cdot D(x_2) = (x_1^d \bmod N) \cdot (x_2^d \bmod N) \equiv (x_1 x_2)^d \mod N \\ \implies & D(x_1) \cdot D(x_2) = D(x_1 x_2). \end{aligned}$$

The same holds for the encryption as well. As a result, if signing is performed on the message without any prior formatting (or hashing), an attacker can forge signatures that are validated by a verification process. This leads to the principle that either the hash of a message must be signed or the message must be properly padded prior to signing. But, some padding schemes have been shown to be weak. We mention one in Section 2.4.2.2 and one may also be interested in to see [34–39].

2.4.1 Factoring N

The most straightforward way to break an asymmetric cryptographic system is to solve its underlying mathematical problem. For the case of RSA, this is factoring N directly. So, this problem is in fact not specific to RSA because the attacker does not benefit from additional information provided by the RSA system. The only goal is to find the prime factors p and q given $N = pq$. The most efficient algorithm currently known for general factorization problem is the (General) Number Field Sieve (NFS) [40] method. For large N, its complexity is estimated as

$$\exp\left((c + o(1)) \ln(N)^{\frac{1}{3}} \ln\ln(N)^{\frac{2}{3}}\right)$$

for some $1 < c < 2$. There is also the Elliptic Curve Method (ECM) [41] that is mostly used to find the small prime factors of a number. This is useful when one wants to check if a number r is m-smooth meaning that all prime factors of r is less than or equal to m. The largest number factored using ECM until now has 83-digits and this record is achieved by Ryan Propper in September 2013.

The largest RSA modulus (without having a prime factor of a special form) factorized in the open literature until now is for $n = 768$ bits [42]. NFS method is used in this process. Today, it is strongly suggested to use RSA with a key size of at least 2048 bits.

2.4.2 Implementation Attacks

This section deals with the attacks that take advantage of the individual implementation characteristics of RSA. Hence, they do not always work and are specific to certain implementations.

2.4.2.1 Side-Channel Analysis

Side-channel analysis is a technique where the attacker tries to exploit weaknesses due to implementation details. For example, consider the *square-and-multiply* algorithm that is used to exponentiate a number. Suppose that we want to calculate $c^d \bmod N$ for some ciphertext c and decryption exponent d (which is n-bit). Let $d = (d_{n-1}, d_{n-2}, \ldots, d_0)$ be the binary representation of d. The algorithm works as follows:

```
1. $A \leftarrow c$ and $B \leftarrow 1$. For $i \in \{0, 1, \ldots, n-1\}$ do:
    (a) if $d_i = 1$, $B \leftarrow A \cdot B \mod N$
    (b) $A \leftarrow A^2 \mod N$
2. return $B$
```

The key point here is that statement (a) is run only when $d_i = 1$, which means a device running this algorithm behaves differently according to the value of d_i. If an attacker can differentiate these situations (i.e., $d_i = 0$ and $d_i = 1$), then he learns the bits of d. Possible ways to observe such a distinguishing property include timing each iteration, examining the power consumption or the noise and etc (e.g., see [43–45]).

Side-channel attacks received a lot of attention after Kocher introduced timing attacks [46]. Simple and differential power analysis [47] is one of the core attacks in the area. A countermeasure for protecting against exposing bits of d is called *exponent blinding*. In this method, one uses $d' = d + a\phi(N)$, $a \in \mathbb{Z}$ instead of d as the decryption exponent. That way an attacker observing the bits of the decryption exponent via a side-channel analysis cannot learn the bits of d due to masking. Another advantage of this method is that decryption may be fastened by decreasing the hamming weight of the decryption exponent.

There is also an important result by Boneh, DeMillo and Lipton [48] that one can expose the prime factors of N by injecting random faults on a device running CRT-RSA [18] signing process. However, this attack may simply be prevented by checking the signature before revealing it.

2.4.2.2 Bleichenbacher's Attack

Bleichenbacher's attack [49] works on an old version of Public Key Cryptography Standard 1 (PKCS 1). According to this standard, a message m is first formatted as follows:

02	Random	00	m

where each value in the first and the third blocks represent a byte. Then, this will be the message to be encrypted. In some protocols, when a ciphertext is decrypted, the application first checks if the most significant 2 bytes are ‘02’ and sends an error message if the check fails [8]. But, now, an attacker can intercept a ciphertext c, generate $c' := rc \bmod N$ for some random value r and send c' to the application. If the application sends an error message, the attacker knows that the most significant 2 bytes of the plaintext is not equal to ‘02’. Thus, the attacker has a mechanism that tells him whether the most significant 16 bits of the decryption of chosen ciphertexts equal to ’02’ or not. Bleichenbacher shows that this mechanism enables an attacker to decrypt c.

2.4.3 Message Recovery Attacks

This section concentrates on RSA attacks that recover the message rather than the private keys. So, the attacker can only find the plaintexts encrypted under certain conditions and cannot completely break the system by obtaining the private keys.

2.4.3.1 Håstad’s Attack

In [50], Håstad shows attacks on the case when the same message m is sent to multiple recipients who use the same public exponent e but different moduli N_i. Let us focus on the easier case first. Let $e = 3$ and Alice send $c_i = E_i(m) = m^3 \bmod N_i$ to 3 recipients, $1 \leq i \leq 3$. If N_i’s are not pairwise relatively prime, then we can find a common factor by a simple greatest common divisor (gcd) computation. Thus, we can assume that N_i’s are pairwise relatively prime. In that case, using the Chinese Remainder Theorem on c_1, c_2 and c_3, an attacker can compute

$$c \equiv m^3 \mod N_1N_2N_3$$

Observe that since $m < N_i$ for all $1 \leq i \leq 3$, $m^3 < N_1N_2N_3$. Thus, the equality $m^3 = c$ is satisfied over $\mathbb{Z}$. The attacker can now easily extract m by computing the real cube root of c.

Håstad [50] proves a stronger result as well. He showed that applying fixed polynomials to m_i before encryption does not prevent the attack provided that the message is encrypted into enough number of ciphertexts. Boneh, in [8], shows a stronger version of this result.

Theorem 2.2 ([50] and [8])**.** *Let $\{f_i\}_{1 \leq i \leq k}$ be a set of polynomials defined over $\mathbb{Z}_{N_i}[x]$ with a degree of at most d where N_i’s are pairwise relatively prime integers. Suppose* *there is a unique m that is less than all N_i satisfying the condition*

$$f_i(m) \equiv 0 \mod N_i, \quad \text{for all } 1 \leq i \leq k.$$

One can recover m efficiently if $k > d$ and (N_i, g_i) pairs are given for $1 \leq i \leq k$.

2.4.3.2 Franklin-Reiter Attack

Franklin-Reiter's attack [51] works when two related messages are encrypted using the same public key pair (N, e) where e is small. For example, let $e = 3$ and $m_1 \neq m_2$ such that $m_1 = f(m_2) \bmod N$ for some affine transformation f. Then,

$$\begin{aligned} c_1 &\equiv m_1^3 \equiv f(m_2)^3 \mod N, \text{ and} \\ c_2 &\equiv m_2^3 \mod N \end{aligned}$$

Thus, setting $h_1 := f(x)^3 - c_1$ and $h_2 := x^3 - c_2$ with $h_1, h_2 \in \mathbb{Z}_N[x]$, we see that $x - m_2$ divides both h_1 and h_2. Thus, by calculating $\gcd(h_1, h_2)$, we can recover this linear factor. In the rare cases of $e \neq 3$, gcd computation may be non-linear and hence the attack fails in these cases.

2.4.3.3 Coppersmith's Short Pad Attack

In this attack [12], Coppersmith considers the case when a message is meant to be sent, it is first appended with a simple random padding and then encrypted as usual. He shows that when this padding is small enough, the message can be recovered without knowing the values of the individual paddings. Suppose m is a message of bitsize at most $n - r$ where $r = \lfloor n/e^2 \rfloor$. Then, the Coppersmith's result shows that m can be obtained efficiently given (N, e) when bitsize of the paddings (which are unknown to the attacker) are at most r.

2.4.4 Attacks Using Extra Knowledge on RSA Parameters

As explained earlier, RSA may be very weak when the parameters satisfy certain conditions. In this section, we describe pioneering works that benefits from some implicit knowledge. Partial key exposure attacks fall into this class as well, but they use some explicit knowledge. They take advantage of some knowledge that the parameters satisfy certain conditions as well as of that some bits of the private exponent d is exposed to the attacker. This exposure may be due the side-channel attacks described earlier in Section 2.4.2.1.

2.4.4.1 Wiener's Attack

In 1990, Wiener introduced an attack [10] on RSA using *continued fraction* techniques. He showed that N can be factored efficiently whenever $d < \frac{1}{3}N^{\frac{1}{4}}$. This attack is a basis for many modern attacks on RSA including the partial key exposure attacks. It actually shows that RSA can be efficiently broken with some implicit knowledge. Before proving this result, we recall some properties from the theory of continued fractions.

Definition 2.3. $[a_0, a_1, \ldots]$ for $a_i \in \mathbb{Z}$ is called the continued fraction representation of $x \in \mathbb{R}$ if

$$x = a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \cdots}}$$

If we stop at an index $i \geq 0$, then $[a_0, a_1, \ldots, a_i]$ is called the i-th convergent of x. The following theorem by Legendre [52] is used for the proof of the attack.

Theorem 2.4. *Let $x \in \mathbb{R}$. If $\frac{a}{b}$ is a fraction satisfying*

$$\left| x - \frac{a}{b} \right| < \frac{1}{2b^2},$$

then $\frac{a}{b}$ is a convergent of x.

Now, we can state and prove the result of Wiener's attack.

Theorem 2.5. *For a given RSA modulus $N = pq$ with $q < p < 2q$, assume that $d < \frac{1}{3}N^{\frac{1}{4}}$. Then, d can be found in time polynomial in n.*

Proof. Recalling the RSA equation, we know that

$$\begin{aligned} & ed = 1 + k\phi(N) \\ \Longrightarrow \quad & \frac{e}{\phi(N)} - \frac{k}{d} = \frac{1}{d\phi(N)} \end{aligned}$$

Also, note that $ed > k\phi(N)$ implies $k < d$ since $e < \phi(N)$. So, by the assumption on d, $k < \frac{1}{3}N^{\frac{1}{4}}$. Now, observe that

$$\begin{aligned}\left|\frac{e}{N} - \frac{k}{d}\right| = \left|\frac{ed - kN}{dN}\right| &= \left|\frac{ed - k\phi(N) + k\phi(N) - kN}{dN}\right| \\ &= \left|\frac{1 - k(N - \phi(N))}{dN}\right|\end{aligned}$$

It is easy to see that $N - \phi(N) = pq - (p-1)(q-1) = p + q - 1 < 3N^{\frac{1}{2}}$. So, we get from the above equalities together with the condition on k that

$$\left|\frac{e}{N} - \frac{k}{d}\right| \leq \left|\frac{3kN^{\frac{1}{2}}}{dN}\right| = \left|\frac{3k}{dN^{\frac{1}{2}}}\right| < \left|\frac{1}{dN^{\frac{1}{4}}}\right| < \left|\frac{1}{3d^2}\right|$$

Hence, using Theorem 2.4, we can find $\frac{k}{d}$ as a convergent of $\frac{e}{N}$. Then, the fact that k and d are relatively prime allows us to find k and d, simultaneously. □

After finding k and d, we can find $\phi(N)$, and hence $s := p + q$. As a result, we can factor N by finding the roots of the polynomial $f(x) = x^2 - sx + N$.

2.4.4.2 Boneh-Durfee Attack

Boneh and Durfee's attack [22] is an important improvement on Wiener's attack. It shows that N can be factored efficiently whenever $d < N^{0.292}$. But, since this attack requires more knowledge about finding the small roots of polynomials, we study this attack in more detail in Section 3.2.3.

Chapter 3

Preliminaries

3.1 Lattice Theory

In this section, we revisit some basic definitions and theorems about lattice theory, which are useful in the partial key exposure attacks. For the sake of completeness, only an introductory part of lattice theory is covered here. For a detailed study on the relation of the Lattice Theory to cryptography, we refer the reader to [53], which also covers some lattice based cryptographic algorithms.

Definition 3.1 (Euclidean Norm for vectors)**.** Let $v = (v_0, \ldots, v_r)$, $r \geq 0$, be a vector in $\mathbb{R}^{r+1}$. Then, the Euclidean Norm of v, denoted by $||v||$, is given as follows:

$$||v|| := \sqrt{\sum_{i=0}^{r} (v_i)^2}.$$

Definition 3.2 (Lattice)**.** Let $v_1, \ldots, v_w \in \mathbb{R}^m$, $w, m \in \mathbb{N}^+$ be a set of $\mathbb{R}$-linearly independent vectors. Then, the lattice L generated by these vectors is defined as

$$L := \{\alpha_1 v_1 + \ldots + \alpha_w v_w \ : \ \alpha_i \in \mathbb{Z} \text{ for } 1 \leq i \leq w\}.$$

So, L is the integer linear combinations of the given vectors.

Remark 3.3. Since given vectors are linearly independent, it must hold that $w \leq m$. If $w = m$, then L is called a *full rank lattice.*

As in the vector spaces, any set of linearly independent vectors generating L is called *a basis* of L, and the dimension of L (denoted by $dim(L)$) is the number of vectors in a basis of L. For the above definition, $dim(L) = w$. L can also be represented by a matrix $\mathcal{M}$ whose row vectors consist of the basis vectors such that

$$\mathcal{M} = \begin{pmatrix} v_1 \\ . \\ . \\ . \\ v_w \end{pmatrix}.$$

The determinant of a lattice L is denoted by $det(L)$. The exact definition of the determinant requires additional definitions, which is not useful for this work (e.g., see [53]). For this work, it is enough to know that for a full rank lattice L, $det(L) = det(\mathcal{M})$.

Remark 3.4. A lattice L admits infinitely many bases, but $det(L)$ is unique.

In this study, we work only with lattices having full rank with the goal of finding small vectors in such lattices. There is an important theorem, due to Hermite, related to this issue (the theorem is attributed to Minkowski as well. e.g., in [9] and [21]). It states in terms of $dim(L)$ and $det(L)$ that a small vector exists in the lattice L. However, it does not describe a constructive method which can be used to find such a small vector.

Theorem 3.5 (Hermite's Theorem)**.** *Let L be a lattice with $dim(L) = w$. Then, there exists a nonzero vector v in L such that*

$$||v|| \leq \sqrt{w}.det(L)^{\frac{1}{w}}.$$

In general, computational complexity of finding a small vector in a lattice increases exponentially in the dimension of the lattice. This problem is NP-hard for randomized reductions [54]. To that end, the lattice reduction algorithm *LLL* introduced by Lenstra, Lenstra and Lovász [15] is used in practice. What LLL does is that given a set of basis vectors for a lattice, it produces a set of *small* vectors describing the same lattice. To see how small the reduced basis vectors are, the following theorem is used:

Theorem 3.6 ([15] and [55]). *For a lattice L with $dim(L) = w$, the LLL algorithm produces a set of reduced basis vectors $\{r_1, \ldots, r_w\}$ such that*

$$||r_i|| \leq 2^{\frac{w(w-1)}{4(w+1-i)}} det(L)^{\frac{1}{w+1-i}}.$$

In particular, for $i = 1$,

$$||r_1|| \leq 2^{\frac{w-1}{4}} det(L)^{\frac{1}{w}}.$$

For a of this theorem, one can see [55]. The computational complexity of the LLL algorithm is polynomial in $dim(L)$ and in the maximal bitsize of an entry [56].

3.2 Finding Small Roots of Polynomials

For a univariate modular polynomial or a multivariate integer polynomial, there is no generic method for finding its roots. Otherwise, RSA would have been easily broken by finding the roots of the RSA equation (Equation 2.1). A novel step forward for achieving this goal is taken by Coppersmith. Coppersmith developed techniques in [12] (see also [13] and [14]) both for finding small *modular* roots of univariate polynomials and also for finding small *integer* roots of bivariate polynomials under certain conditions. Before we start explaining the method, let us give some definitions and notations.

As usual, we denote $\mathbb{Z}_N := \mathbb{Z}/N\mathbb{Z}$. A multivariate (or univariate) polynomial with coefficients from $\mathbb{Z}$ is denoted by f. If the coefficients of a polynomial are in $\mathbb{Z}_N$, then the polynomial is represented by f_N. Below, the definitions are given for f, but those for f_N are analogous.

Definition 3.7 (Monomial). Let $f(x_1, \ldots, x_s)$, $s \geq 1$, be a polynomial. Then, any term $x_1^{i_1} \ldots x_s^{i_s}$, $i_1, \ldots, i_s \in \mathbb{N}$, with a nonzero coefficient is called a monomial.

Definition 3.8 (Leading Monomial and Leading Coefficient). Leading monomial of a multivariate polynomial is the largest monomial with respect to an ordering. Also, the coefficient of the leading monomial is called the leading coefficient.

Definition 3.9 (Euclidean Norm for polynomials). Let $f(x_1, \ldots, x_s)$, $s \geq 1$, be a polynomial whose coefficients are represented by $a_0, \ldots, a_t$ for some $t \in \mathbb{N}$. Then, the Euclidean

Norm of f, denoted by $||f||$, is given as

$$||f|| := \sqrt{\sum_{i=0}^{t} (a_i)^2}.$$

Now, we are ready to formally define the problems that are solved using Coppersmith's methods:

- given $X \in \mathbb{Z}^+$, find a root x_0 of $f_N(x)$ modulo N such that $|x_0| < X$ where the factorization of N is unknown.
- given $X, Y \in \mathbb{Z}^+$, find a root (x_0, y_0) of $f(x, y)$ such that $|x_0| < X$ and $|y_0| < Y$ where the factorization of N is unknown.

In this work, we study the extended version of the first problem of finding small modular root of univariate polynomials to multivariate polynomials. The second problem of finding small integer roots requires more careful work due to an independence issue that is discussed in Chapter 5 in detail.

3.2.1 Finding Small Modular Roots

Assume that $f_N(x)$ has root x_0 modulo N. The key idea behind Coppersmith's method is to find a polynomial $h(x)$ that has the same root x_0 over integers (not modulo N). But how can we find this polynomial $h(x)$? To do that, we construct polynomials having the root x_0 modulo N^m for some $m > 0$. These polynomials represent a lattice and we find a small vector in this lattice using LLL algorithm. And, finally, we hope that the polynomials corresponding to these small vectors carry the root x_0 over $\mathbb{Z}$. From now on, we let s denote the number of variables in a multivariate polynomial and w the dimension of the lattice in an attack.

Our aim now is to explain this idea more formally. Let $\epsilon > 0$, and for $0 \leq k \leq m$ and some j construct a set of polynomials g_{jk} as follows:

$$g_{jk}(x) := x^j (f_N(x))^k N^{m-k}.$$

These polynomials are called *shift polynomials*. It is easy to see that all shift polynomials share the root x_0 modulo N^m. That's why it must be the case that any integer linear combination $h(x)$ of g_{jk}'s must have the root x_0 modulo N^m. If it also satisfies the condition that $|h(x_0)| < N^m$, then $h(x_0) = 0$ over $\mathbb{Z}$. Hence, we obtain the end result.

Having made these observations, Howgrave-Graham's following theorems [57] (which are reformulations of Coppersmith's ideas) sum up the conditions as follows.

Theorem 3.10 (Howgrave-Graham's Theorem for univariate case, [57]). *Let $f(x) \in \mathbb{Z}[x]$, be a polynomial with the number of monomials less than or equal to w. If the following two conditions hold:*

1. *$f(x_0) \equiv 0 \bmod M$ for some $|x_0| < X$ and an $M \in \mathbb{Z}^+$,*
2. *$||f(xX)|| < \frac{M}{\sqrt{w}}$,*

then x_0 is a root of f over $\mathbb{Z}$.
i.e., $f(x_0) = 0$.

Proof. Let $f(x) := a_0 + a_1x + \cdots + a_mx^m$ for some $m \in \mathbb{N}$. Note that since f has at most w monomials, the set of nonzero a_i's can have at most w elements. First, by the definition of the Euclidean Norm, we have

$$\frac{M}{\sqrt{w}} > ||f(xX)|| = \sqrt{\sum_{i=0}^{m}(a_iX^i)^2}.$$

And, also using triangle inequality yields to

$$\begin{aligned}|f(x)| = |a_0 + a_1x + \cdots + a_mx^m| \leq & |a_0| + |a_1x| + \cdots + |a_mx^m| \\ \leq & |a_0| + |a_1X| + \cdots + |a_mX^m|.\end{aligned}$$

Finally, by Hölder's inequality, we get

$$\sum_{i=0}^{m}|a_iX^i| \leq \sqrt{w}.\sqrt{\sum_{i=0}^{m}(a_iX^i)^2} = \sqrt{w}.||f(xX)|| < M$$

So, $|f(x_0)| < M$ and by the first assumption, $f(x_0) \equiv 0 \bmod M$. This yields $f(x_0) = 0$ over $\mathbb{Z}$. □

A similar proof may be found in [21] as well. This theorem can also be generalized for multivariate polynomials as given below. The proof for the multivariate case works analogously to the one for the univariate case. Only notations get a bit more complicated.

Theorem 3.11 (Howgrave-Graham's Theorem for multivariate case, [57]).
Let $f(x_1, \ldots, x_s) \in \mathbb{Z}[x_1, \ldots, x_s]$, $s \geq 1$, be a polynomial with the number of monomials less than or equal to w. If the following two conditions hold:

1. $f(x_1^0, \ldots, x_s^0) \equiv 0 \bmod M$ *for some* $|x_1^0| < X_1, \ldots, |x_s^0| < X_s$ *and an* $M \in \mathbb{Z}^+$,
2. $||f(x_1X_1, \ldots, x_sX_s)|| < \frac{M}{\sqrt{w}}$,

then $(x_1^0, \ldots, x_s^0)$ is a root of f over $\mathbb{Z}$.
i.e., $f(x_1^0, \ldots, x_s^0) = 0$.

Let $f_N(x)$ be the polynomial whose root x_0 we are trying to extract. Suppose also that we have constructed w shift polynomials g_{jk}'s. We see the coefficients of $g_{jk}(xX)$ as vectors forming a basis for a lattice L (note that $dim(L) = w$). We apply LLL to these vectors and get reduced vectors corresponding to the reduced polynomials $r_1(xX), \ldots, r_w(xX)$. By Theorem 3.6, we know that

$$||r_1(xX)|| \leq 2^{\frac{w-1}{4}} det(L)^{\frac{1}{w}}.$$

If it is also the case that

$$2^{\frac{w-1}{4}} det(L)^{\frac{1}{w}} < \frac{M}{\sqrt{w}},$$

then Howgrave-Graham's theorem is satisfied and $h(x) = r_1(x)$ can be chosen. As a result, the main goal reduces to choosing g_{jk}'s so that $det(L)$ is small.

The case for multivariate polynomials requires more work. Let $f_N(x_1, \ldots, x_s)$ be the polynomial whose root $(x_1^0, \ldots, x_s^0)$ we are trying to extract. Similarly, after building shift polynomials as

$$g_{j_1 \cdots j_s k}(x_1, \ldots, x_s) := x_1^{j_1} \cdots x_s^{j_s} (f_N(x_1, \ldots, x_s))^k N^{m-k},$$

construct a lattice L using the coefficient vectors of $g_{j_1 \cdots j_s k}(x_1X_1, \ldots, x_sX_s)$'s. Applying LLL to L, we get reduced polynomials $r_1(x_1X_1, \ldots, x_sX_s), \ldots, r_w(x_1X_1, \ldots, x_sX_s)$

where $w \geq s$. By Theorem 3.6, we know that

$$||r_1(x_1X_1, \ldots, x_sX_s)|| \leq \cdots \leq ||r_s(x_1X_1, \ldots, x_sX_s)|| \leq 2^{\frac{w(w-1)}{4(w+1-s)}} det(L)^{\frac{1}{w+1-s}}.$$

If it is also the case that

$$2^{\frac{w(w-1)}{4(w+1-s)}} det(L)^{\frac{1}{w+1-s}} < \frac{M}{\sqrt{w}}, \tag{3.1}$$

then Howgrave-Graham's theorem for multivariate polynomials is satisfied and we get s polynomials carrying the root $(x_1^0, \ldots, x_s^0)$ over $\mathbb{Z}$. Now, to extract the common root, there are mainly two techniques that we can use: computing resultants or a Gröbner basis. In our experiments, a Gröbner basis computation is more efficient and resultant computation consume too much memory. Moreover, we mostly found more than s polynomials carrying the common root. i.e., the system of equations is overdefined, from which a Gröbner basis computation benefits. However, in order for a Gröbner basis computation to find the common root, the following heuristic assumption needs to hold.

Assumption 1. Let $f_1, \cdots, f_k$ be the polynomials having the desired root over $\mathbb{Z}$ for $k \geq s$ computed using LLL reduction. Furthermore, let I be the ideal generated by these polynomials. Then, the algebraic variety of I is zero-dimensional. In particular, the common root can be extracted by computing a Gröbner basis on I.

Since our attack in Chapter 4 relies on this assumption, it is heuristic. However, our experiments show that this assumption holds in general (see Section 4.3). The computational complexity of a Gröbner basis computation can be bounded by a polynomial in $\log N$ assuming the number of variables and the maximal degree of input polynomials is fixed [58].

As in similar works, we simplify the condition (3.1) to

$$det(L) < M^w \tag{3.2}$$

and let low order terms contribute to an error term ϵ. Thus, in our attack, the ultimate goal is to construct a lattice satisfying Inequality 3.2. Next, the computational complexity of the overall attack is discussed.

3.2.2 Complexity of the Attacks

Partial key exposure attacks using Coppersmith's method consist mainly of two parts: polynomial reduction, which is usually done using LLL, and root extraction, which can be done by computing resultants or a Gröbner basis. In the following, computational complexities of these operations are discussed in more detail. In the following results, we assume that the error term ϵ introduced in the methods for choosing shift polynomials and the number of variables s are fixed.

3.2.2.1 Polynomial Reduction

Today one of the fastest LLL reduction algorithms (i.e., having the lowest worstcase complexity) is suggested by Nguyen and Stehlé [56]. It is called *floating-point LLL (fpLLL)* (which is implemented in Sage and Magma [59]). The complexity of the algorithm is $O(w^5(w + \eta)\eta)$ where w is the dimension of the lattice and η is the maximal bitsize of an entry. In Coppersmith methods, η is polynomial in n and w depends only on ϵ. Hence, the polynomial reduction phase of the Coppersmith's method can be done in time polynomial in n using the LLL reduction algorithm.

3.2.2.2 Root Extraction

As mentioned before, there are two general ways to extract a common root from multivariate polynomials: resultant or Gröbner basis computations. To compute a Gröbner basis for root extraction, F5 algorithm by Faugère [58] can be used. The complexity of the algorithm can be bounded by a polynomial in n assuming that the number of variables and the maximal degree of input polynomials is fixed.

One may wish to use the method of resultants. Computing a resultant is equivalent to computing the determinant of a matrix. For a square matrix of size ℓ, the determinant computation can be done in time $O(\ell^3)$. Note that, in the case of Coppersmith's method, the number of iterations for the resultant computations is fixed (remember that s is fixed), the number of terms in a reduced polynomial is constant and the size of the coefficients in each resultant polynomial is polynomial in the coefficient sizes of the starting polynomials (whose sizes are polynomial in n). Hence, the total complexity of the resultants computation step can be upper bounded by a polynomial in n.

As a result, partial key exposure attacks using Coppersmith's method can be done in time polynomial in n assuming that ϵ and s are fixed.

3.2.3 Boneh-Durfee Attack

Now that we have discussed the method of finding small roots of polynomials, we can illustrate Boneh and Durfee's attack [22].

Theorem 3.12 ([22]). *Let $N = pq$ be of bitsize n where p, q are primes with $q < p < 2q$. Also, let $ed \equiv 1 \bmod \phi(N)$ with $e \sim N$ and $d \sim N^{\beta}$. Then, if $d < N^{0.292}$, then under Assumption 1, N can be factored in time polynomial in n.*

Proof. Recalling the RSA equation, we know that

$$ed - 1 = k(N - (p + q - 1)).$$

Looking at this equation modulo e and replacing k with x and $p + q - 1$ with y, Boneh and Durfee use the following polynomial:

$$f_e(x, y) = 1 + xN - xy.$$

It is easy to see that f_e carries the root $(x_0, y_0) := (k, p+q-1)$ modulo e. We also know that $x_0 < \min\{e, d\} = \min\{N, N^{\beta}\} = N^{\beta}$ and $y_0 < 3N^{\frac{1}{2}}$. Hence, the upperbounds become $X = N^{\beta}$ and $Y = N^{\frac{1}{2}}$ ignoring small constants. Now, the shift polynomials are defined as

$$\begin{aligned} g_{ik}(x, y) &= x^i f_e^k(x, y) e^{m-k}, \qquad \text{for } i = 0, \dots, m - k \\ h_{jk}(x, y) &= y^j f_e^k(x, y) e^{m-k}, \qquad \text{for } j = 0, \dots, t \end{aligned}$$

and for $k = 0, 1, \dots, m$. Setting $t = \tau m$, Inequality (3.2) becomes

$$X^{\frac{1}{6}m^3(2+3\tau)+o(m^3)} Y^{\frac{1}{6}m^3(1+3\tau+3\tau^2)+o(m^3)} < N^{\frac{1}{6}m^3(1+3\tau)+o(m^3)}.$$

To get an asymptotic bound, we ignore $o(m^3)$ terms. Substituting the values for X and Y, the condition reduces to

$$\begin{aligned} & \beta(2+3\tau)+\frac{1}{2}(1+3\tau+3\tau^2) < 1+3\tau \\ \Longrightarrow \quad & 3\tau^2+3\tau(2\beta-1)+(4\beta-1) < 0. \end{aligned} \tag{3.3}$$

which reaches its maximum at $\tau = \frac{1}{2} - \beta$. Plugging in this value to Inequality 3.3, we finally obtain

$$\begin{aligned} -3\beta^2 + 7\beta - \frac{7}{4} < 0 & \\ \Longrightarrow \quad \beta < \frac{7}{6} - \frac{1}{3}\sqrt{7} \approx 0.284. & \end{aligned}$$

This result is then improved by looking at the sublattices of the lattice generated by the shift polynomials. Boneh and Durfee's final result is given as $\beta < 0.292$ [22]. □

Chapter 4

Partial Key Exposure Attacks on Multi-Power RSA

In this chapter, we describe known attacks on multi-power RSA that uses Coppersmith's methods along with a new partial key exposure attack on multi-power RSA when the exponents are generated modulo $\phi(N)$. Section 4.2 is based on [29], which has been accepted to CAI 2015 (joint work with the supervisor and the co-supervisor).

Notation: Let log denote the logarithm base 2 unless the base is given concretely. We use the following notation throughout this chapter.

N	Multi-power RSA modulus
n	bitsize of N
p, q	prime factors of N
r	integer satisfying the relation $N = p^r q$
e	RSA public exponent
d	RSA private exponent
d_0	known part of d
$\tilde{d}$	unknown part of d
α	$\log_N e$ (i.e., $e \approx N^\alpha$)
β	$\log_N d$ (i.e., $d \approx N^\beta$)
δ	$\log_N d_0$ (i.e., $d_0 \approx N^\delta$)
γ	$\log_N \tilde{d}$ (i.e., $\tilde{d} \approx N^\gamma$)

Partial key exposure attacks uses Coppersmith's method of finding small roots of polynomials [12] (see Section 3.2). An attacker obtains some bits of d (e.g., via side-channel attacks, see Section 2.4.2.1) and tries to construct all of d. The more realistic scenario (and the one most partial key exposure attacks focus on) is that the attacker knows some consecutive most significant bits (MSBs) or consecutive least significant bits (LSBs) of it. A partial key exposure attack taking advantage of MSBs (resp. LSBs) is referred as a *known MSBs (resp. LSBs) attack*.

4.1 Known Attacks

As there are two methods of generating exponents e and d for multi-power RSA, we study each case separately.

4.1.1 Attacks when $ed \equiv 1 \bmod (p-1)(q-1)$

Less work has been done for this variant of multi-power RSA. In 2008, Itoh et al. [23] showed a small private exponent attack such that N could be factored efficiently when $d < N^{\frac{2-\sqrt{2}}{r+1}}$. Later, same ideas in [23] are used in [28] to mount partial key exposure attacks. It is shown that N can be factored efficiently when

$$\begin{array}{rrcll} & \gamma & \leq & \frac{7}{4(r+1)} - \frac{1}{4}\sqrt{\frac{24(\alpha+\beta)}{r+1} - \frac{39}{(r+1)^2}} & \text{if MSBs or middle bits are known,} \\ \text{or} & \gamma & \leq & \frac{5}{3(r+1)} - \frac{2}{3}\sqrt{\frac{3(\alpha+\beta)}{r+1} - \frac{5}{(r+1)^2}} & \text{if LSBs are known.} \end{array}$$

Note that these attacks do not work when e and d are of full size modulo $(p-1)(q-1)$ (i.e., $e, d \approx N^{\frac{2}{r+1}}$). We argue the validity of the known MSBs attack in [28] in Chapter 5.

4.1.2 Attacks when $ed \equiv 1 \bmod (p^r - p^{r-1})(q-1)$

To begin with, we discuss small private exponent attacks that do not require any knowledge of the bits of d. The first one is described by Takagi in [24] where he introduced the multi-power RSA variant. In this paper, Takagi shows that N can be factored in time polynomial in n when

$$\beta < \frac{1}{2(r+1)}.$$

Later, this bound is improved by May [25] such that

$$\beta < \max\left\{\frac{r}{(r+1)^2}, \frac{(r-1)^2}{(r+1)^2}\right\}.$$

Sarkar [26] improved this bound even further for $r \leq 5$. His result is not given explicitly because the function is complicated but rather he presents numerical values (see Table 4.1). More recent work is done by Lu et al. [27]. They improve the bound of Sarkar when $r \geq 4$. Their result is given explicitly as $\beta < \frac{r(r-1)}{(r+1)^2}$. In fact, their attack works whenever the unknown part $\tilde{d}$ of d (whether it is all of d or an MSB/LSB part of it) satisfies $\tilde{d} < N^{\frac{r(r-1)}{(r+1)^2}}$. This directly implies partial key exposure attacks. Their partial key exposure attacks are independent of the sizes of e and d.

In Table 4.1, we compare numerical values of the aforementioned bounds on the size of d. We would like to note that for large r, multi-power RSA variant is not useful for practical purposes since the prime factors of N get smaller for a fixed-sized N. Thus, N can be factored more easily. Moreover, the result of [60] shows that N can be factored efficiently when $\frac{1}{r+1}$ fraction of the bits of p are known.

Table 4.1: Numerical comparison between the bounds on β.

	bound on β			
r	Takagi	May	Sarkar	Lu et al.
2	0.166	0.222	**0.395**	0.222
3	0.125	0.250	**0.410**	0.375
4	0.100	0.360	0.437	**0.480**
5	0.083	0.444	0.464	**0.555**
6	0.071	0.510	0.489	**0.612**

Lu et al.'s partial key exposure attacks require slightly less knowledge of the bits of d than our new attack when $e > N^{0.846}$ or $d > N^{0.962}$ where the other exponent is full-sized. Otherwise, our attack is better and requires much less knowledge about d as e or d gets smaller. One should note that Takagi's decryption process is efficient only when e is small as described in [24]. Hence, small-e attacks have more practical interest.

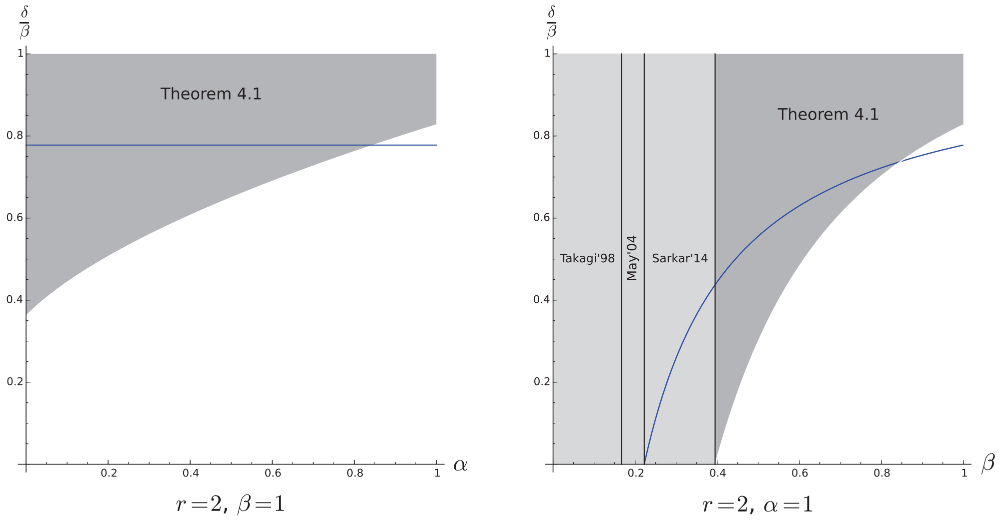

FIGURE 4.1: The relation between the sizes of e (resp. d) and the fraction of the part of d required to be known.

4.2 A New Attack with Known LSBs

The attack basically uses partial knowledge of LSBs and works for all e (resp. d) when the exponent d (resp. e) has full size bit length.[1] More concretely, we prove the following theorem which generalizes Sarkar's result [26].

Theorem 4.1. *Let $r \geq 2$ be an integer and $N = p^r q$ be a multi-power RSA modulus, where p and q are distinct primes with the same bit size (i.e., $p, q \approx N^{\frac{1}{r+1}}$). Suppose that $ed \equiv 1 \bmod \phi(N)$ with $e \approx N^{\alpha}$ and $d \approx N^{\beta}$. Suppose further that an attacker obtains an LSB part d_0 of d, where $d_0 \geq N^{\delta}$ for some $\delta \in \mathbb{R}^{\geq 0}$. Then under Assumption 1, there exists an algorithm which finds the prime factors of N in polynomial time in* $\log N$ *provided that*

$$\rho(r, \beta, \alpha, \delta) < 0,$$

where ρ is a function of r, β, α and δ.

In Figure 4.1, we show the attack regions of previous works along with our result for the case $r = 2$. Light grey areas indicated by "Takagi'98", "May'04", "Sarkar'14" shows the attack regions by [24], [25] and [26], respectively. The blue curves are the fixed bound of [27] (i.e., $\tilde{d} \approx N^{\frac{2}{9}}$). The darker grey areas are the applicable regions of our attack.

[1] This rule is induced by the condition that $ed \equiv 1 \bmod \phi(N)$.

Since our result in Theorem 4.1 relies on Assumption 1, it is heuristic. However, our experiments show that this assumption holds in general (see Section 4.3). Now, we can proceed with the proof of our main result Theorem 4.1.

Proof. (Theorem 4.1) Multi-power RSA parameters satisfy the congruence $ed \equiv 1 \bmod \phi(N)$ with $\phi(N) = (p^r - p^{r-1})(q-1)$. This implies the equation that $ed - 1 = k(p^r - p^{r-1})(q-1)$ for some $k \in \mathbb{Z}$. Since we know an LSB part of d, we can write this as $eM\tilde{d} + ed_0 - 1 = k(p^r - p^{r-1})(q-1)$ where $d = \tilde{d}M + d_0$ and M is a power of 2. Hence, we have the following polynomial

$$f_{eM}(x, y, z) = ed_0 - 1 - xN - xy^{r-1} + xy^{r-1}z + xy^r$$

carrying the root $(x_0, y_0, z_0) = (k, p, q)$ modulo eM. It is easy to see that $|x_0| < X := N^{\alpha+\beta-1}$, $|y_0| < Y := N^{\frac{1}{r+1}}$ and $|z_0| < Z := N^{\frac{1}{r+1}}$ neglecting small constants.

Let $m, t_1, t_2 \geq 0$ and define the following shift polynomials:

$$\begin{aligned} g_{i,j,k}(x, y, z) &= x^j y^k z^{j+t_1} f^i_{eM}(x, y, z), \\ &\quad \text{where } i = 0, \cdots, m,\ j = 1, \cdots, m-i \text{ and } k = j, \cdots, j + 2r - 2, \\ g_{i,0,k}(x, y, z) &= y^k z^{t_1} f^i_{eM}(x, y, z), \\ &\quad \text{where } i = 0, \cdots, m \text{ and } k = 0, \cdots, t_2. \end{aligned}$$

Recall that $y_0^r z_0 = N$. Hence, we replace every occurrence of $y^r z$ with N in the shift polynomials. Denote new polynomials by $g'_{i,j,k}(x, y, z)$. Observe that choosing xy^r as the leading monomial of f_{eM}, the leading monomials in $g'_{i,j,k}$'s are of the form $x^{i+j}y^{k+ri-rl}z^{j+t_1-l}$, where $l = \min\left\{\lfloor \frac{k+ri}{r} \rfloor, j + t_1\right\}$.

Let a_ℓ denote the leading coefficient. Assuming $\gcd(a_\ell, eM) = 1$, we can multiply $g'_{i,j,k}$'s with the inverse a'_ℓ of their corresponding leading coefficient in $\mathbb{Z}/(eM)^m\mathbb{Z}$. Finally, the shift polynomials become

$$h_{i,j,k}(x, y, z) = a'_\ell \cdot g'_{i,j,k}(x, y, z) \cdot (eM)^{m-i}$$

which carry the root (x_0, y_0, z_0) modulo $(eM)^m$.

We let the coefficient vectors of $h_{i,j,k}(xX, yY, zZ)$ represent the basis vectors of a lattice L. Generation of L is summarized in Algorithm 1.

Algorithm 1 Generating the Lattice L

```
Input: $r \geq 2$; $m, t_1, t_2 \geq 0$ and $f_{eM}(x, y, z)$

$G, H, Ord \leftarrow \emptyset$
for $i \in \{0, 1, \cdots, m\}$ do
    for $j \in \{1, 2, \cdots, m - i\}$ do
        for $k \in \{j, j + 1, \cdots, j + 2r - 2\}$ do
            Append $(x^j y^k z^{j+t_1} f_{eM}^i, i)$ to $G$
            $l \leftarrow \min\left\{\lfloor \frac{k+ri}{r} \rfloor, j + t_1\right\}$
            Append $x^{i+j} y^{k+ri-rl} z^{j+t_1-l}$ to $Ord$
        end for
    end for
end for
for $i \in \{0, 1, \cdots, m\}$ do
    for $k \in \{0, 1, \cdots, t_2\}$ do
        Append $(y^k z^{j+t_1} f_{eM}^i, i)$ to $G$
        $l \leftarrow \min\left\{\lfloor \frac{k+ri}{r} \rfloor, j + t_1\right\}$
        Append $x^{i+j} y^{k+ri-rl} z^{j+t_1-l}$ to $Ord$
    end for
end for
for each element $(g, i)$ in $G$ do
    Replace each occurrence of $y^r z$ with $N$ in $g$
    $a'_\ell \leftarrow a_\ell^{-1} \bmod eM$, where $a_\ell$ is the leading coefficient of $g$
    Append $(a'_\ell \cdot g \cdot (eM)^{m-i})$ to $H$
end for
$i \leftarrow 1$
for each polynomial $h(x, y, z)$ in $H$ do
    Set $i$-th row of $L$ to the coefficient vector of $h(xX, yY, zZ)$ ordered w.r.t. $Ord$
    Increment $i$
end for
```

Note that each polynomial in H generated by Algorithm 1 introduces exactly one new monomial, which is appended to Ord that defines the monomial ordering. Hence, the matrix representing the lattice is lower triangular when each row is ordered with respect to Ord. As a result, the determinant of L is the product of the diagonal entries of the representation matrix.

$$\begin{aligned} det(L) = & \left(\prod_{i=0}^{m} \prod_{j=1}^{m-i} \prod_{k=j}^{j+2r-2} X^{i+j} Y^{k+ri-rl_1} Z^{j+t_1-l_1} (eM)^{m-i} \right) \\ & \times \left(\prod_{i=0}^{m} \prod_{k=0}^{t_2} X^{i} Y^{k+ri-rl_2} Z^{t_1-l_2} (eM)^{m-i} \right), \end{aligned}$$

where $l_1 = \min\left\{ \lfloor \frac{k+ri}{r} \rfloor, j+a \right\}$ and $l_2 = \min\left\{ \lfloor \frac{k+ri}{r} \rfloor, a \right\}$. Letting s_x, s_y, s_z and s_{eM} be the powers of X, Y, Z and eM in $det(L)$, respectively, and denoting the dimension of the lattice by w, we obtain

$$w = \sum_{i=0}^{m} \sum_{j=1}^{m-i} \sum_{k=j}^{j+2r-2} 1 + \sum_{i=0}^{m} \sum_{k=0}^{t_2} 1 = \frac{2r-1}{2} m^2 + t_2 m + o(m^2),$$

$$s_x = \sum_{i=0}^{m} \sum_{j=1}^{m-i} (2r-1)(i+j) + \sum_{i=0}^{m} \sum_{k=0}^{t_2} i = \frac{2r-1}{3} m^3 + \frac{t}{2} m^2 + o(m^3),$$

$$s_{eM} = \sum_{i=0}^{m} \sum_{j=1}^{m-i} (2r-1)(m-i) + \sum_{i=0}^{m} \sum_{k=0}^{t_2} (m-i) = \frac{2r-1}{3} m^3 + \frac{t}{2} m^2 + o(m^3).$$

Assuming $\frac{t_2}{r} \leq t_1 \leq m$, we obtain as an asymptotic result

$$\begin{aligned}
s_y &= \sum_{i=0}^{m} \sum_{j=1}^{m-i} \sum_{k=j}^{j+2r-2} (k + ri - rl_1) + \sum_{i=0}^{m} \sum_{k=0}^{t_2} (k + ri - rl_2) \\
&\approx \sum_{i=t_1}^{\lfloor \frac{(r-1)m+rt_1}{2r-1} \rfloor} \sum_{j=1}^{\lfloor \frac{r(i-t_1)}{r-1} \rfloor} \sum_{k=j}^{j+2r-2} (k + ri - rj - rt_1) \\
&+ \sum_{i=\lfloor \frac{(r-1)m+rt_1}{2r-1} \rfloor}^{m} \sum_{j=1}^{m-i} \sum_{k=j}^{j+2r-2} (k + ri - rj - rt_1) \\
&+ \sum_{i=t_1-\lfloor \frac{t_2}{r} \rfloor}^{t_1} \sum_{k=r(t_1-i)}^{t_2} (k + ri - rt_1) + \sum_{i=t_1}^{m} \sum_{k=0}^{t_2} (k + ri - rt_1) \\
&= \frac{1}{2} \left(\frac{r^2 m^3}{3} - r^2 m^2 t_1 + r^2 m t_1^2 - \frac{r^2 t_1^3}{3} + r m^2 t_2 \right. \\
&\qquad \left. -2rmt_1t_2 + rt_1^2 t_2 + mt_2^2 - t_1 t_2^2 + \frac{t_2^3}{3r} \right) + o(m^3),
\end{aligned}$$

$$\begin{aligned}
s_z &= \sum_{i=0}^{m} \sum_{j=1}^{m-i} \sum_{k=j}^{j+2r-2} (j + t_1 - l_1) + \sum_{i=0}^{m} \sum_{k=0}^{t_2} (t_1 - l_2) \\
&\approx \sum_{i=0}^{t_1} \sum_{j=1}^{m-i} \sum_{k=j}^{j+2r-2} \left(j + t_1 - \frac{k+ri}{r} \right) + \sum_{i=t_1}^{\lfloor \frac{(r-1)m+rt_1}{2r-1} \rfloor} \sum_{j=\lfloor \frac{r(i-t_1)}{r-1} \rfloor}^{m-i} \sum_{k=j}^{j+2r-2} \left(j + t_1 - \frac{k+ri}{r} \right) \\
&+ \sum_{i=0}^{t_1-\lfloor \frac{t_2}{r} \rfloor} \sum_{k=0}^{t_2} \left(t_1 - \frac{k+ri}{r} \right) + \sum_{i=t_1-\lfloor \frac{t_2}{r} \rfloor}^{t_1} \sum_{k=0}^{r(t_1-i)} \left(t_1 - \frac{k+ri}{r} \right) \\
&= \frac{1}{2} \left(\frac{(r-1)^2 m^3}{3r} + (r-1)^2 m^2 t_1 + rmt_1^2 - \frac{rt_1^3}{3} + t_1^2 t_2 - \frac{t_1 t_2^2}{r} + \frac{t_2^3}{3r^2} \right) + o(m^3),
\end{aligned}$$

which are approximated as in [26].

Note that the simplified condition Inequality 3.2 is $det(L) < (eM)^{wm}$. In our case, we need

$$s_x(\alpha + \beta - 1) + (s_y + s_z) \left(\frac{1}{r+1} \right) + (s_{eM} - wm)(\alpha + \delta) < 0$$

to be satisfied. Plugging in the values for s_x, s_y, s_z and s_{eM}, we obtain a polynomial $\rho'(r, \alpha, \beta, \delta)$ with parameters t_1, t_2 and m. Let $t_1 = \tau_1 m$ and $t_2 = \tau_2 m$, and terms of $o(m^3)$ contribute to an error term ϵ. Next, we take the partial derivative of ρ' with respect to τ_1 and τ_2, and find the values making the derivatives zero to obtain the maximum value of ρ'. Finally, for $\gamma := \beta - \delta$, when $\tau_1 = \frac{1-r\gamma+r^2(1-\gamma)}{2r}$ and

$$\tau_2 = \frac{1 + r^3(1-\gamma) - r^2(1+2\gamma) + r(1-\gamma) + 2r\sqrt{r^2(1-\gamma) + r(1-2\gamma) + 1 - \gamma}}{2r+2}$$

both derivatives become zero. Plugging in these values in ρ', a function $\rho(r, \alpha, \beta, \delta)$. When the tuple $(r, \alpha, \beta, \delta)$ satisfies $\rho(r, \alpha, \beta, \delta) < 0$, Howgrave-Graham's theorem follows. We can extract the root (k, p, q) under Assumption 1, and thus factor N in time polynomial in $\log N$. □

Remark 4.2. We note that our definition of shift polynomials is similar to the one in [26]. The difference is that we work modulo eM instead of modulo e. Hence, the constant coefficient of f_{eM} changes.

Corollary 4.3. *Equating $M = 1$ (i.e., $\delta = 0$), we obtain the result of Sarkar [26] as a corollary of Theorem 4.1 where no knowledge about the bits of d is required.*

TABLE 4.2: Numerical values satisfying $\rho < 0$ for different r and α values where $\beta = 1$.

r	smallest δ value satisfying $\rho(r) < 0$ for $\alpha = 1$	smallest δ value satisfying $\rho(r) < 0$ for $\alpha = 0$
2	0.828	0.362
3	0.798	0.344
4	0.750	0.314
5	0.703	0.285
6	0.662	0.259
7	0.625	0.237

Unfortunately, the exact expression of ρ is complicated. Thus, in Table 4.2, we provide some numerical values for δ which yields $\rho < 0$ when β is fixed to 1. We remind that for $r = 2$ new attack regions are given in Table 4.1 when either d or e is full-sized.

4.3 Experimental Results

In this section, we provide various experimental results. In all of our experiments, we fix d to be full-sized (i.e., $\beta = 1$) which is mostly the case in real-life applications. The values for p, q and d are chosen randomly (or d is the inverse of $2^{16} + 1$ modulo $\phi(N)$). The experiments are performed on Sage 6.5 running on Ubuntu 14.04 LTS with Intel Core i7-3770 CPU at 3.40GHz and 16GB RAM.

Our results are given in Table 4.3 and Table 4.4. In all of our experiments, Gröbner basis computation yields a polynomial of the form $y-p$ giving the factorization of N. For the case when $\alpha = \beta = 1$ (which is illustrated in Table 4.3), we would like to highlight that our result in some examples is better than the theoretical bound $\delta \geq 0.828$. However, when e is chosen small (e.g., $e = 2^{16}+1$), the modulus eM becomes very small when compared to the case $\alpha = \beta = 1$. Therefore, the low order terms ignored to simplify the condition to $det(L) < (eM)^{wm}$ have much higher effect in this case. Thus, the results are a little bit worse than the best possible bound of Theorem 4.1.

Table 4.3: Experimental results for $\alpha = \beta = 1$. $n = 2048$ bits for the last row and $n = 1024$ bits for the rest.

r	m	t_1	t_2	w	δ	LLL time (secs)	Gröbner Basis time (secs)
2	6	4	7	119	0.870	1930.21	3.00
2	7	4	8	156	0.860	6517.26	67.99
2	8	4	7	180	0.850	19619.96	1227.18
2	8	5	9	198	0.835	28684.34	358.80
2	9	5	9	235	0.830	63748.97	635.33
2	9	5	10	245	**0.823**	67480.18	149.56
3	7	4	9	220	0.952	26671.68	7358.66
2	8	5	9	198	0.840	90981.76	2246.77

Table 4.4: Experimental results for $e = 2^{16}+1$, $\beta = 1$. $n = 2048$ bits for the last two rows and $n = 1024$ bits for the rest.

r	m	t_1	t_2	w	δ	LLL time (secs)	Gröbner Basis time (secs)
2	8	3	2	135	0.520	21234.57	4114.00
2	8	3	2	135	0.510	19082.57	4280.77
2	9	3	3	175	0.500	48950.79	9134.06
2	10	3	2	198	0.485	84090.70	15927.35
3	9	3	3	265	0.510	148030.34	56230.82
2	10	3	2	198	0.500	203293.58	45573.57
2	10	3	2	198	0.490	185964.22	40817.77

We present a numerical example as well. Let $N = p^2q$ for

$$\begin{aligned} p = {} & 4851381352638300223177429586853498256903097882540048 \\ & 5254486419527749771974248071032894272105256038885213, \\ q = {} & 5367862136571034162844637067723677810733386174980963 \\ & 8967939577192028487545707276530594101760884089716 27. \end{aligned}$$

Also, let e, d be as follows

$$\begin{aligned}
e = \; & 86472774791379904562530989436203277712168664790504314\\
& 6682311792067772911802157359816012044809156666501181\\
& 321356983959223368812264279920170244935605591454813\\
& 9745237274009032763052680918659976522485946082137 18\\
& 0817924915034967412817180354624477408110754521853 32\\
& 6434547344350843585566869376119365889231256441643 27,\\
d = \; & 58143026404937222258552087067167738089154589779014286\\
& 7420294368674221211121589293844375961073911127523 11\\
& 2160337584097074208157067789519440156374784116880 55\\
& 4024001358562445057484343311006703646988566193204 57\\
& 5270306085125085676515116201772953762909645649161 89\\
& 1711097508729004754201118653384561209836938819100 07.
\end{aligned}$$

Then, when 845 LSBs of d are given ($\delta = 0.823$), p, q can be found using our attack. In this case, N is 1024-bits, and e, d are both 1023-bits.

Chapter 5

Conclusion and Discussions

In this thesis, we study multi-power RSA where $N = p^r q$ for $r \geq 2$. First, we revisit the RSA cryptosystem and some of its cryptanalytic aspects. Later, Coppersmith's method of finding small roots of polynomials is discussed [12]. Using these preliminaries, we show a new partial key exposure attack on multi-power RSA that takes advantage of known LSBs. Our result in Theorem 4.1 generalizes the work of Sarkar [26]. Moreover, we provide experimental results justifying our claims. Our attack works even in the case when $e, d \approx N$. Furthermore, our attack has the advantage that it requires less knowledge about d when e is small. This has more practical interest because in most of the practical applications, $e = 2^{16} + 1$ is chosen for efficiency reasons. Recall that e must be small in order for multi-power RSA to be efficient [24].

One may wonder why our attack is not directly applicable to known MSBs case. Suppose that we know an MSB part d_0 of d. Then, we obtain the equation

$$ed_0 + e\tilde{d} - 1 = k(p^r - p^{r-1})(q - 1),$$

where $\tilde{d}$ represents the unknown part of d. Considering this equation as a polynomial, we get

$$F(w, x, y, z) = 1 - ed_0 - ew + x(N - y^r - y^{r-1}z + y^{r-1}).$$

Now e, N or ed_0 are possible choices of moduli. The case e is studied in [26] where one cannot benefit from partial knowledge of d as it vanishes. If N is chosen as the modulus, then the trick of replacing each term $y^r z$ with N and finding its inverse cannot be applied. That leaves us with the option to choose ed_0 as the modulus. This case actually corresponds to finding a small root of *integer* equations [14], not modular equations [13].

Observe that reducing F modulo ed_0 does not eliminate any variable. In particular, F_{ed_0} and F have the same set of monomials. Hence, the polynomials derived from LLL may just be those of the form $F \cdot g_i$ for nonzero polynomials g_i not carrying the desired root. More concretely, the attacker does not obtain any additional useful information at all although LLL-reduced polynomials carry the root since they have the factor F.

For a recent work, one may see Coron's works [61, 62] about methods to ensure independence between the initial polynomial F and the polynomials derived after LLL reduction (this independence is also ensured in Coppersmith's method [14]). Unfortunately, the tricks used in this work cannot be directly applied together with Coron's method.

This issue raises questions about the validity of known MSBs attack shown in [28]. The authors do not specify any methodology guaranteeing the independence aforementioned. Their experiments for this case are very far away from the new attack region described by Theorem 1 in their paper. Moreover, the authors also state that in some experiments, they just verified that the LLL-reduced polynomials contain the root. As we explained earlier, this does not have any implication for an attacker to be able to find the root.

In some cases, we successfully show experimental results that are better than the theoretical bound. This may happen due to the fact that LLL algorithm does not use all the rows in the representation matrix and concentrates on a submatrix. Thus, a further study option may be investigating sublattices of the original lattice to improve the theoretical bound. Actually, this is first done in [22] by introducing *geometrically progressive matrices*. However, this is a hard task because the lattice will not be of full rank in this case and calculating the determinant gets complicated. Thus, one needs to develop new strategies for parametric calculation of the determinant of the lattice.

One clear extension of attacks on multi-power RSA could be investigating attacks when the modulus has the form $N = p^r q^s$ for $r, s \geq 1$. If one can show a result for any r and s that reaches the best known bounds on different RSA variants, the designers can have a better idea on how to balance between efficiency and security (for a study on this balance problem, see [63]). The final step would be to combine all these attacks by considering

$N = p^{r_1}p^{r_2}\cdots p^{r_s}$ where $s \geq 2$ and $r_i \geq 1$ for $1 \leq i \leq s$. Very recently, a method for factoring $N = p^r q^s$ for large r and s is studied in [64].

Another further study option could be to improve upon the works of Coron [61, 62] for finding integer roots of polynomials. As mentioned, the tricks used in this work (and also in similar works) are not directly applicable with Coron's methods. Finding a more flexible strategy for ensuring the independence discussed earlier would result in better known MSBs attacks on different RSA variants (including the standard RSA).

The most crucial part of attacks using Coppersmith's methods is the lattice reduction. Currently, the LLL algorithm is used for this purpose. If one can find a new efficient algorithm for finding short vectors in a lattice or improve the LLL algorithm, this would have great implications in not only the attacks discussed in this work but also in various areas in Mathematics.

Bibliography

[1] D. R. Stinson. *Cryptography: Theory and Practice.* CRC Press, Inc., Boca Raton, FL, USA, 1st edition, 1995.

[2] W. Diffie and M. Hellman. New directions in cryptography. *IEEE Trans. Inf. Theor.*, 22(6):644–654, 1976.

[3] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, 1978.

[4] T. Elgamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *Information Theory, IEEE Transactions on*, 31(4):469–472, Jul 1985.

[5] I. F. Blake, G. Seroussi, and N. Smart. *Elliptic curves in cryptography*, volume 265. Cambridge university press, 1999.

[6] I. Blake, G. Seroussi, N. Smart, and J. W. S. Cassels. *Advances in Elliptic Curve Cryptography (London Mathematical Society Lecture Note Series).* Cambridge University Press, New York, NY, USA, 2005.

[7] E. Barker, W. Barker, W. Burr, W. Polk, and M. Smid. Recommendation for key management - part 1: general. National Institue of Standards and Technology, NIST Special Publication 800 - 57 Part 1 Revision 3, July 2012. URL `http://csrc.nist.gov/publications/nistpubs/800-57/sp800-57_part1_rev3_general.pdf`. Accessed: 2015-05-09.

[8] D. Boneh. Twenty years of attacks on the RSA cryptosystem. *Notices of the AMS*, 46:203–213, 1999.

[9] M. J. Hinek. *Cryptanalysis of RSA and Its Variants.* Chapman & Hall/CRC, 1st edition, 2009.

[10] M. J. Wiener. Cryptanalysis of short RSA secret exponents. *IEEE Transactions on Information Theory*, 36:553–558, 1990.

[11] D. Boneh, G. Durfee, and Y. Frankel. An attack on RSA given a small fraction of the private key bits. In Kazuo Ohta and Dingyi Pei, editors, *Advances in Cryptology - ASIACRYPT '98*, volume 1514 of *Lecture Notes in Computer Science*, pages 25–34. Springer Berlin Heidelberg, 1998.

[12] D. Coppersmith. Small solutions to polynomial equations, and low exponent RSA vulnerabilities. *Journal of Cryptology*, 10(4):233–260, 1997.

[13] D. Coppersmith. Finding a small root of a univariate modular equation. In Ueli Maurer, editor, *Advances in Cryptology - EUROCRYPT '96*, volume 1070 of *Lecture Notes in Computer Science*, pages 155–165. Springer Berlin Heidelberg, 1996.

[14] D. Coppersmith. Finding a small root of a bivariate integer equation; factoring with high bits known. In Ueli Maurer, editor, *Advances in Cryptology - EUROCRYPT '96*, volume 1070 of *Lecture Notes in Computer Science*, pages 178–189. Springer Berlin Heidelberg, 1996.

[15] A. K. Lenstra, H. W. Lenstra, and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515–534, 1982.

[16] J. Blömer and A. May. New partial key exposure attacks on RSA. In Dan Boneh, editor, *Advances in Cryptology - CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 27–43. Springer Berlin Heidelberg, 2003.

[17] M. Ernst, E. Jochemsz, A. May, and B. de Weger. Partial key exposure attacks on RSA up to full size exponents. In Ronald Cramer, editor, *Advances in Cryptology - EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 371–386. Springer Berlin Heidelberg, 2005.

[18] J.-J. Quisquater and C. Couvreur. Fast decipherment algorithm for RSA public-key cryptosystem. *Electronics Letters*, 18(21):905–907, October 1982.

[19] M. Joye and T. Lepoint. Partial key exposure on RSA with private exponents larger than N. In Mark D. Ryan, Ben Smyth, and Guilin Wang, editors, *Information Security Practice and Experience*, volume 7232 of *Lecture Notes in Computer Science*, pages 369–380. Springer Berlin Heidelberg, 2012.

[20] E. Jochemsz and A. May. A strategy for finding roots of multivariate polynomials with new applications in attacking RSA variants. In Xuejia Lai and Kefei Chen, editors, *Advances in Cryptology - ASIACRYPT 2006*, volume 4284 of *Lecture Notes in Computer Science*, pages 267–282. Springer Berlin Heidelberg, 2006.

[21] E. Jochemsz. *Cryptanalysis of RSA Variants Using Small Roots of Polynomials*. PhD thesis, Technische Universiteit Eindhoven, 2007.

[22] D. Boneh and G. Durfee. Cryptanalysis of RSA with private key d less than $N^{0.292}$. *Information Theory, IEEE Transactions on*, 46(4):1339–1349, Jul 2000.

[23] K. Itoh, N. Kunihiro, and K. Kurosawa. Small secret key attack on a variant of RSA (due to Takagi). In Tal Malkin, editor, *Topics in Cryptology - CT-RSA 2008*, volume 4964 of *Lecture Notes in Computer Science*, pages 387–406. Springer Berlin Heidelberg, 2008.

[24] T. Takagi. Fast RSA-type cryptosystem modulo $p^k q$. In Hugo Krawczyk, editor, *Advances in Cryptology - CRYPTO '98*, volume 1462 of *Lecture Notes in Computer Science*, pages 318–326. Springer Berlin Heidelberg, 1998.

[25] A. May. Secret exponent attacks on RSA-type schemes with moduli $N = p^r q$. In Feng Bao, Robert Deng, and Jianying Zhou, editors, *Public Key Cryptography - PKC 2004*, volume 2947 of *Lecture Notes in Computer Science*, pages 218–230. Springer Berlin Heidelberg, 2004.

[26] S. Sarkar. Small secret exponent attack on RSA variant with modulus $N = p^r q$. *Designs, Codes and Cryptography*, 73(2):383–392, 2014. ISSN 0925-1022.

[27] Y. Lu, R. Zhang, and D. Lin. New results on solving linear equations modulo unknown divisors and its applications. Cryptology ePrint Archive, Report 2014/343, 2014. `http://eprint.iacr.org/`.

[28] Z. Huang, L. Hu, J. Xu, L. Peng, and Y. Xie. Partial key exposure attacks on Takagi's variant of RSA. In Ioana Boureanu, Philippe Owesarski, and Serge Vaudenay, editors, *Applied Cryptography and Network Security*, volume 8479 of *Lecture Notes in Computer Science*, pages 134–150. Springer International Publishing, 2014.

[29] M. F. Esgin, M. S. Kiraz, and O. Uzunkol. A new partial key exposure attack on multi-power RSA. In *Algebraic Informatics*, Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2015. (To appear).

[30] T. M. Apostol. *Introduction to analytic number theory*, volume 1. Springer Science & Business Media, 1976.

[31] A. K. Lenstra, J. P. Hughes, M. Augier, J. W. Bos, T. Kleinjung, and C. Wachter. Ron was wrong, whit is right. Cryptology ePrint Archive, Report 2012/064, 2012. `http://eprint.iacr.org/`.

[32] N. Heninger, Z. Durumeric, E. Wustrow, and J. A. Halderman. Mining your Ps and Qs: Detection of widespread weak keys in network devices. In *Proceedings of the 21st USENIX Conference on Security Symposium*, Security'12, pages 35–35, Berkeley, CA, USA, 2012. USENIX Association.

[33] D. J. Bernstein, Y.-A. Chang, C.-M. Cheng, L.-P. Chou, N. Heninger, T. Lange, and N. van Someren. Factoring RSA keys from certified smart cards: Coppersmith in the wild. In Kazue Sako and Palash Sarkar, editors, *Advances in Cryptology - ASIACRYPT 2013*, volume 8270 of *Lecture Notes in Computer Science*, pages 341–360. Springer Berlin Heidelberg, 2013.

[34] Y. Desmedt and A. M. Odlyzko. A chosen text attack on the RSA cryptosystem and some discrete logarithm schemes. In *Advances in Cryptology*, CRYPTO '85, pages 516–522, London, UK, 1986. Springer-Verlag.

[35] J.-S. Coron, D. Naccache, and J. P. Stern. On the security of RSA padding. In Michael Wiener, editor, *Advances in Cryptology - CRYPTO '99*, volume 1666 of *Lecture Notes in Computer Science*, pages 1–18. Springer Berlin Heidelberg, 1999.

[36] F. Grieu. A chosen messages attack on the ISO/IEC 9796-1 signature scheme. In Bart Preneel, editor, *Advances in Cryptology - EUROCRYPT 2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 70–80. Springer Berlin Heidelberg, 2000.

[37] E. Brier, C. Clavier, J.-S. Coron, and D. Naccache. Cryptanalysis of RSA signatures with fixed-pattern padding. In *Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '01, pages 433–439, London, UK, 2001. Springer-Verlag.

[38] A. K. Lenstra and I. E. Shparlinski. Selective forgery of RSA signatures with fixed-pattern padding. In David Naccache and Pascal Paillier, editors, *Public Key Cryptography*, volume 2274 of *Lecture Notes in Computer Science*, pages 228–236. Springer Berlin Heidelberg, 2002.

[39] J.-S. Coron, D. Naccache, M. Tibouchi, and R.-P. Weinmann. Practical cryptanalysis of ISO/IEC 9796-2 and EMV signatures. In Shai Halevi, editor, *Advances in Cryptology - CRYPTO 2009*, volume 5677 of *Lecture Notes in Computer Science*, pages 428–444. Springer Berlin Heidelberg, 2009.

[40] A. K. Lenstra and H. W. Lenstra, editors. *The development of the number field sieve*, volume 1554 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin, 1993.

[41] H. W. Lenstra. Factoring integers with elliptic curves. *Annals of Mathematics*, 126 (3):pp. 649–673, 1987.

[42] T. Kleinjung, K. Aoki, J. Franke, A. K. Lenstra, E. Thomé, J. W. Bos, P. Gaudry, A. Kruppa, P. L. Montgomery, D. A. Osvik, H. te Riele, A. Timofeev, and P. Zimmermann. Factorization of a 768-bit RSA modulus. In Tal Rabin, editor, *Advances in Cryptology - CRYPTO 2010*, volume 6223 of *Lecture Notes in Computer Science*, pages 333–350. Springer Berlin Heidelberg, 2010.

[43] B. den Boer, K. Lemke, and G. Wicke. A DPA attack against the modular reduction within a CRT implementation of RSA. In *Cryptographic Hardware and Embedded Systems - CHES 2002*, volume 2523 of *Lecture Notes in Computer Science*, pages 228–243. Springer Berlin Heidelberg, 2003.

[44] D. Genkin, A. Shamir, and E. Tromer. RSA key extraction via low-bandwidth acoustic cryptanalysis. In *Advances in Cryptology - CRYPTO 2014*, volume 8616 of *Lecture Notes in Computer Science*, pages 444–461. Springer Berlin Heidelberg, 2014.

[45] D. Genkin, L. Pachmanov, I. Pipman, and E. Tromer. Stealing keys from pcs using a radio: Cheap electromagnetic attacks on windowed exponentiation. Cryptology ePrint Archive, Report 2015/170, 2015. `http://eprint.iacr.org/`.

[46] P. C. Kocher. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In Neal Koblitz, editor, *Advances in Cryptology - CRYPTO '96*,

volume 1109 of *Lecture Notes in Computer Science*, pages 104–113. Springer Berlin Heidelberg, 1996.

[47] P. C. Kocher, J. Jaffe, and B. Jun. Differential power analysis. In *Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '99, pages 388–397, London, UK, UK, 1999. Springer-Verlag.

[48] D. Boneh, R. A. DeMillo, and R. J. Lipton. On the importance of checking cryptographic protocols for faults. In *Proceedings of the 16th Annual International Conference on Theory and Application of Cryptographic Techniques*, EUROCRYPT '97, pages 37–51, Berlin, Heidelberg, 1997. Springer-Verlag.

[49] D. Bleichenbacher. Chosen ciphertext attacks against protocols based on the RSA encryption standard PKCS #1. In *Proceedings of the 18th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '98, pages 1–12, London, UK, UK, 1998. Springer-Verlag.

[50] J. Håstad. Solving simultaneous modular equations of low degree. *SIAM Journal on Computing*, 17(2):336–341, 1988.

[51] D. Coppersmith, M. Franklin, J. Patarin, and M. Reiter. Low-exponent RSA with related messages. In Ueli Maurer, editor, *Advances in Cryptology - EUROCRYPT '96*, volume 1070 of *Lecture Notes in Computer Science*, pages 1–9. Springer Berlin Heidelberg, 1996.

[52] A. M. Legendre. *Essai sur la théorie des nombres.* Duprat, Paris, 1798.

[53] J. Hoffstein, J. Pipher, and J. H. Silverman. *An Introduction to Mathematical Cryptography.* Springer Publishing Company, Incorporated, 1 edition, 2008.

[54] M. Ajtai. The shortest vector problem in L_2 is NP-hard for randomized reductions (extended abstract). In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*, STOC '98, pages 10–19, New York, NY, USA, 1998. ACM.

[55] A. May. *New RSA Vulnerabilities Using Lattice Reduction Methods.* PhD thesis, University Paderborn, 2003.

[56] P. Q. Nguyen and D. Stehlé. Floating-Point LLL revisited. In Ronald Cramer, editor, *Advances in Cryptology - EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 215–233. Springer Berlin Heidelberg, 2005.

[57] N. Howgrave-Graham. Finding small roots of univariate modular equations revisited. In Michael Darnell, editor, *Crytography and Coding*, volume 1355 of *Lecture Notes in Computer Science*, pages 131–142. Springer Berlin Heidelberg, 1997.

[58] J. C. Faugère. A new efficient algorithm for computing Gröbner Bases without reduction to zero (F5). In *Proceedings of the 2002 International Symposium on Symbolic and Algebraic Computation*, ISSAC '02, pages 75–83, New York, NY, USA, 2002. ACM.

[59] W. Bosma, J. Cannon, and C. Playoust. The Magma algebra system. I. The user language. *J. Symbolic Comput.*, 24(3-4):235–265, 1997. Computational algebra and number theory (London, 1993).

[60] D. Boneh, G. Durfee, and N. Howgrave-Graham. Factoring $n = p^r q$ for large r. In *Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '99, pages 326–337, London, UK, UK, 1999. Springer-Verlag.

[61] J.-S. Coron. Finding small roots of bivariate integer polynomial equations revisited. In Christian Cachin and Jan L. Camenisch, editors, *Advances in Cryptology - EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 492–505. Springer Berlin Heidelberg, 2004.

[62] J.-S. Coron. Finding small roots of bivariate integer polynomial equations: A direct approach. In Alfred Menezes, editor, *Advances in Cryptology - CRYPTO 2007*, volume 4622 of *Lecture Notes in Computer Science*, pages 379–394. Springer Berlin Heidelberg, 2007.

[63] S. D. Galbraith, C. Heneghan, and J. F. McKee. Tunable balancing of RSA. In *Proceedings of the 10th Australasian Conference on Information Security and Privacy*, ACISP'05, pages 280–292, Berlin, Heidelberg, 2005. Springer-Verlag.

[64] J.-S. Coron, J.-C. Faugère, G. Renault, and R. Zeitoun. Factoring $n = p^r q^s$ for large r and s. Cryptology ePrint Archive, Report 2015/071, 2015. `http://eprint.iacr.org/`.