Cryptography

Samuel Jaques and Christophe Petit

Trinity Term 2022

Contents

1 Introduction

The course will cover introductory and advanced topics in cryptology. This year, the advanced part will focus on cryptanalysis (by far the most popular option based on our poll), and more specifically on cryptanalysis techniques related to factoring and RSA.

Learning outcomes: The students will be able to manipulate security definitions and proofs by reductions as used in cryptography. They will understand the main cryptographic tools in use today, the security guarantees they can provide and their limitations. They will also learn some advanced and research topics in cryptography.

Synopsis This course will be organized as a reading course. We will assume no prior exposition to cryptology, hence start with basic concepts including essential cryptographic primitives (digital signatures, public and private key encryption, hash functions), proofs by reduction and major cryptographic algorithms in use today. We will then cover some factoring algorithms including the number field sieve, and finally techniques to break RSA without factoring (classically).

Organization The course is divided in four parts. For each part:

- We provide a set of reference reading and questions to guide you through this reading.
- We also provide a group project made of a few additional questions. This typically includes both theoretical questions and some implementation work.
- We ask you to prepare a 30min-1h oral presentation with your answers.
- We will meet every other week of the term for 2 hours. We will start these meetings with your presentation, often interrupting it with questions, then carry on with any further question you may have on the reading material.

Advanced warning Christophe Petit and Sam Jaques will be colecturing this module, with Christophe expected to cover the first and last part while Sam covers the middle parts.

There is however a non negligible probability that plans are changed with last minute notice as Christophe is expecting a child on May 7th. Apologies in advance for any inconvenience this might cause.

Reading list

- Katz-Lindell, Introduction to Modern Cryptography [6].
- Galbraith, Mathematics of Public Key Cryptography [4].

Forum: There is a forum for the course on the Moodle page, intended for discussions and questions about the readings, exercises, group projects, and any other course-related topics. Since we will meet infrequently, please use the forum to ask questions about the material as you read through it.

2 Symmetric key cryptography

The main reference for this part is Katz-Lindell's book "Introduction to Modern Cryptography" [6].

2.1 Individual guided reading

Introduction Reference: Katz-Lindell [6], Chapter 1.

1. What is an encryption scheme? Get acquainted with the following terminology: plaintext, ciphertext, key, key generation algorithm, encryption algorithm, decryption algorithm.
2. What is Kerckhoff's principle? Do you think this principle makes sense? Could there be sensible exceptions?
3. What is Statistical cryptanalysis ? Implement (in your favorite language) one of the "historical ciphers" discussed in Section 1.3, and a statistical analysis attack against it.
4. Which properties should a "secure" encryption scheme satisfy? Try to come up with your own definition of what this means, then compare this definition with p20-21.
5. Imagine you have designed a new encryption scheme. How could you proceed to prove or argue about its security?
6. Compare the meaning of "Security proof" in cryptography with the usual meaning of "proof" in mathematics. Can a cryptographic scheme with a "security proof" still be insecure? Are "security proofs" valuable in cryptography? Should a "security proof" rather be called "security argument"?

Perfectly secure encryption Reference: Katz-Lindell [6], Chapter 2.

1. Study the "one-time pad" cryptosystem. Design and implement a similar system for symbols in an arbitrary group G.
2. Try to break the one-time pad cryptosystem using statistical analysis. Is it possible?
3. What is the meaning of "perfect secrecy"?
4. Prove that the one-time pad cryptosystem is "perfectly secret".
5. Could you use the one-time pad to make online purchases? Discuss.
6. Give an attack against one-time pad when part of the key is used more than once.

Private key encryption and pseudorandomness Reference: Katz-Lindell [6], Chapter 3.

1. Compare "information theoretical security" and "computational security".
2. Define "efficient" algorithm and "negligible" probability. Compare "asymptotic" and "concrete" usual meanings for these words.
3. Study the indistinguishability definition (Definition 3.9). What does it say? Informally, does it imply that keys remain secret, that the encryption scheme cannot be inverted (without the secret key)? Is it a good security definition of an encryption scheme, or does it fail to capture some attacks that might be realistic?
4. What is a pseudo-random number generator (PRNG)? How can such an object be used in cryptography? What are the security properties that a PRNG must satisfy?
5. The one-time pad is perfectly secure but only if the key used is as large as the message. Describe how a PRNG allow can help remove that assumption. Are the security guarantees provided by the new scheme equivalent?
6. What is a stream cipher and how can such a thing be constructed?
7. Study the definition of IND-CPA security. How does it compare with the previous definition? Does the attack scenario captured by the definition make any sense in practice? Does the definition capture all the attack scenarios you can think of?
8. Suppose that an encryption scheme is deterministic. Show that it cannot satisfy the IND-CPA security property.
9. Study the definition of pseudorandom function (PRF), and how this object is used in Construction 3.25 to build an IND-CPA secure encryption scheme. Try to prove security of the encryption scheme (by reduction to the security of the pseudo-random function).
10. Study the definition of IND-CCA security. How does it compare with the previous definition? Does the attack scenario captured by the definition make any sense in practice? Does the definition capture all the attack scenarios you can think of?
11. Study the definition of pseudorandom permutation (PRP, Definition 3.38) and compare it with that of a pseudorandom function. Relate PRP and block cipher.
12. Does a PRP immediately lead to a secure encryption scheme?
13. What are the main " encryption modes of operations"? What are their respective advantages, and which ones are most secure? Use Google to identify concrete applications where these modes are used.
14. Consider the use of *initial values* (IV) in these constructions. Is it important that these values are chosen randomly?
15. Compare block cipher and stream cipher.

Block cipher design Reference: Katz-Lindell [6], Chapter 5.

1. Try to come up with a sensible bloc cipher design then read Chapter 5. Are there any similarities between your ideas and the ones described there? Will your scheme likely be vulnerable to some of the attacks described?

2. Suppose you need to use a block cipher. Is it best to design a new one, or use an existing one? What if you are an expert on block cipher design?

3. Given a pseudorandom function, how can we (provably) build a pseudorandom permutation? Study the answer provided by Feistel networks.

Message authentication and collision-resistant hash functions Reference: Katz-Lindell [6], Chapter 4.

1. What do we mean by authentication?

2. Is encrypting a message with an IND-CCA encryption scheme sufficient to guarantee message integrity?

3. What is a *message authentication code* (MAC), and how is it formally defined? What are the security properties expected of a MAC?

4. What are replay attacks? Do MACs offer any protection against replay attacks, and why?

5. Review the list of cryptographic objects defined so far, and try to build a MAC from one of them. Can you prove the security of your construction?

6. Study construction 4.3. Can you prove its security? Understand the proof provided in Katz-Lindell.

7. Consider a variable-length MAC constructed as follows: choose a pseudorandom function f; parse the message into blocks m_i of size equal to domain size of f; compute $t_i = f_k(m_i \oplus i)$; compute $t = t_1 \oplus t_2 \oplus \ldots \oplus t_n$. Provide a concrete attack against this construction.

8. What is CBC-MAC? Try to attack CBC-MAC with an attack as the one developed for the previous question, then try to prove its security. Check the security proof provided in the reference book.

9. Show that adding a random IV in the CBC-MAC construction would NOT lead to a secure MAC, i.e. it would lead to an attack against the scheme.

10. What is a message extension attack? Show that CBC-MAC, as described in Construction 4.7, is vulnerable to such an attack. Discuss possible ways to thwart the attack.

11. Define a hash function, collision resistance, second preimage resistance and preimage resistance. What are the relationships between these security properties?

12. What is a random oracle? Relate this notion to the previous ones.

13. Suppose there are randomly chosen 25 people in a room. What is the probability that two of them have the same birthday? How many people are needed for this probability to be larger than 0.5?

14. How hard is it to compute collisions for the most secure of all hash functions?

15. What is the purpose of the Merkle-Damgaard transform? Study Construction 4.11 and discuss its security.

16. Should hash functions be *keyed* or not? Is the key supposed to be secret in this context?

17. Study the NMAC and HMAC constructions and their security proofs.

18. A MAC can also be used to improve an IND-CCA encryption scheme into a CCA-secure encryption scheme. Study Construction 4.17; try to build a CCA attack against it; then try to argue its security; and finally check the proof provided in the book.

19. Suppose you have a secure encryption scheme and a secure MAC. How should you combine them to obtain both the necessary confidentiality and authentication guarantees? Discuss the Enc-and-MAC, Enc-then-MAC, MAC-then-Enc approaches. Under which conditions is each of these approaches secure? Have they been or are they still used in practice?

20. Authenticated encryption

2.2 Group project

The goals of this group project are to practice security reductions and to understand some aspects of pseudorandom generation in cryptography.

1. Formally define pseudo-random number generator, one-way function and hard-core bit.

2. Show how to build a secure PRNG from any one-way function.

3. Linear Feedback Shift Registers (LFSR) were once considered good pseudo-random number generators. Describe what is an LFSR and implement one using your favourite computer language.

4. Explain how the Berlekamp-Massey algorithm can "break" such a construction, and use it to break your own LFSR implementation.

5. Consider the use of *initial values* (IV) in encryption modes of operations. Is it important that these values are chosen randomly? Implement a concrete attack against one mode of operation (with a block cipher of your choice) when your LFSR is used to generate IV values.

Hints: there is a well-known PRNG construction from any one-way function, which you can find in the Katz-Lindell book [6]. There are numerous sources for the Berlekamp-Massey algorithm; a cryptography source is Antoine Joux's book on *Algorithmic Cryptanalysis* [5]. If you cannot implement the algorithm, we suggest that you use an existing implementation available online, with proper referencing.

3 Public key cryptography

3.1 Individual guided reading

Number Theory Background. Chapter 9 from Katz-Lindell, skipping sections 9.2.2, 9.2.5, 9.3.4, and 9.4.

1. (Problem 9.10) For the group $\mathbb{Z}_{24}$:
   (a) List the elements of the group.
   (b) Is this group cyclic?
   (c) Is 18 a generator? Is 5 a generator?
2. (Problem 9.11) For the group $\mathbb{Z}_{21}^*$:
   (a) List the elements in this group. How many are there?
   (b) What is $\phi(21)$?
   (c) Find a generator of this group, and an element that is not a generator.
   (d) More generally: Let p be a prime, and suppose $g \in \mathbb{Z}_p^*$. How do we decide if g generates $\mathbb{Z}_p^*$? How can we *find* a generator g?
3. What is the group structure of $\mathbb{Z}_p^*$? Given this, for which integers m can you compute an mth root modulo p?
4. What does it mean for a group problem to be "easy" or "hard"? In $\mathbb{Z}_p^*$, which operations are easy or hard?
   (a) Finding $z \equiv xy \mod p$, given x and y
   (b) Finding $z \equiv x^2 \mod p$, given x
   (c) Finding z such that $zx \equiv 1 \mod p$, given x
   (d) Finding $z \equiv x^y \mod p$ given x and y with $y < p$
   (e) Finding z such that $x^z \equiv y \mod p$, given x and y
   (f) Finding z such that $z^x \equiv y \mod p$, given x and y
5. PRIMES is the problem of deciding whether an input integer N is prime or not. Easy: show that PRIMES is in co-NP. Hard: show that PRIMES is in NP.
6. Consider the following probabilistic "algorithm" to solve PRIMES:
   (a) On any input n, return "Composite"

   By the prime number theorem, the fraction of integers of size $\approx n$ which are prime is only $O(\frac{1}{\log n})$. This means the algorithm succeeds with probability $1 - O(\frac{1}{\log n})$, which asymptotically approaches 1. Thus, this is a valid probabilistic algorithm for this problem. What is the error in the logic here? How would you define "probabilistic algorithm" to exclude this error?
7. Consider the following experiment $\mathsf{SquareRoot}_{\mathcal{A},\mathsf{GenModulus}}(n)$:

(a) Run $\mathsf{GenModulus}(1^n)$ to obtain (N, p, q).

(b) Select a random $x \in \mathbb{Z}_N^*$

(c) $\mathcal{A}$ is given N and x and outputs y.

(d) The output of the experiment is 1 if $y^2 \equiv x \mod N$, and 0 otherwise.

Show that, given an efficient algorithm $\mathcal{A}$ that succeeds at $\mathsf{SquareRoot}$, there is an efficient algorithm that succeeds at Factor (defined in Section 9.2.3). (This reduction becomes relevant for factoring cryptanalysis later).

8. (Problem 9.19) Formally define the CDH assumption. Prove that hardness of the CDH problem relative to G implies hardness of the discrete-logarithm problem relative to G, and that hardness of the DDH problem relative to G implies hardness of the CDH problem relative to G.

(extra questions that would be good: 9.18 from Katz-Lindell.

Public Key Encryption Reference: Katz-Lindell, Chapter 11, and Chapter 1 (sections 12.1 to 12.4.2, and 12.5.1, 12.5.2, 12.5.4, and 12.5.6)

1. (Problem 11.3) Describe a man-in-the-middle attack on the Diffie–Hellman protocol where the adversary shares a key k_A with Alice and a (different) key k_B with Bob, and Alice and Bob cannot detect that anything is wrong.

2. (Problem 11.4) Consider the following key-exchange protocol:

   (a) Alice chooses uniform $k, r \in \{0,1\}^n$, and sends $s := k \oplus r$ to Bob.

   (b) Bob chooses uniform $t \in \{0,1\}^n$, and sends $u := s \oplus t$ to Alice.

   (c) Alice computes $w := u \oplus r$ and sends w to Bob.

   (d) Alice outputs k and Bob outputs $w \oplus t$.

   Show that Alice and Bob output the same key. Analyze the security of this protocol against a passive eavesdropper.

3. Suppose that the modulus p generated for Diffie-Hellman or El-Gamal is actually composite. What problems will this cause?

4. Pick your favourite cyclic group that isn't $\mathbb{Z}_n^*$ or points on an elliptic curve. Probably this group will not work well for Diffie-Helman or El-Gamal; what property is it missing?

5. Some implementations of RSA use a fixed exponent e (often 3 or 65537). Since new primes p and q are generated each time it is used, there is some risk that e divides $p-1$ or $q-1$. Suppose that this happens, and someone encrypts a (padded) message as $c \equiv m^e \mod N$. Can c be decrypted by someone knowing the secret key (p, q)?

6. Related, suppose someone pads their message m to m', and m' is no longer co-prime to N. What happens when m' is encrypted? Should an implementation check to ensure that this does not occur?

7. Suppose someone re-uses the same value of k for Schnorr signatures. Describe an attack on this.

8. (Problem 12.12):One of the attacks on plain RSA discussed in Section 12.5.1 involves a sender who encrypts two related messages using the same public key. Formulate an appropriate definition of security ruling out such attacks, and show that any CPA-secure public-key encryption scheme satisfies your definition.

9. (Problem 12.14): Consider the following modified version of padded RSA encryption: Assume messages to be encrypted have length exactly $\|N\|/2$. To encrypt, first compute $\hat{m} := \mathsf{0x00}\|r\|\mathsf{0x00}\|m$ where r is a uniform string of length $\|N\|/2 - 16$. Then compute the ciphertext $c := \hat{m}^e \bmod N$. When decrypting a ciphertext c, the receiver computes $\hat{m} := c^d \bmod N$ and returns an error if $\hat{m}$ does not consist of $\mathsf{0x00}$ followed by $\|N\|/2 - 16$ arbitrary bits followed by $\mathsf{0x00}$. Show that this scheme is not CCA-secure. Why is it easier to construct a chosen-ciphertext attack on this scheme than on PKCS #1 v1.5?

Digital Signatures Reference: Chapter 13, up to the end of 13.6 (section 13.5.3 has some material on elliptic curves we can skip). Section 13.7 on TLS is a good extra section if you are interested in the real-world applications.

1. Compare a digital signature scheme to a MAC. In what scenarios would you use each?

2. (Problem 13.2) In Section 13.4.1 we showed an attack on the plain RSA signature scheme in which an attacker forges a signature on an arbitrary message using two signing queries. Show how an attacker can forge a signature on an arbitrary message using a single signing query.

3. (Problem 13.3)Assume the RSA problem is hard. Show that the plain RSA signature scheme satisfies the following weak definition of security: an attacker is given the public key $\langle N, e\rangle$ and a uniform message $m \in \mathbb{Z}_N^*$. The adversary succeeds if it can output a valid signature on m without making any signing queries.

4. (Problem 13.4) Consider a "padded RSA" signature scheme where the public key is $\langle N, e\rangle$ as usual, and a signature on a message $m \in \{0,1\}^\ell$ is computed by choosing uniform $r \in \{0,1\}^{2n-\ell-1}$ and outputting $[(r\|m)^d \bmod N]$.

   (a) How can verification be done for this scheme?

   (b) Show that this scheme is insecure.

5. Last month (April 2022) a lot of zero-knowledge proofs based on the Fiat-Shamir transform were found to have a major vulnerability. In fact this exact vulnerability appears in Construction 13.12 in the Katz-Lindell book. In the interactive Schnorr identification scheme, the verifier has $\mathbb{G}$, q, g, and y (q is the order of g, and $y = g^x$ for the prover's secret key x). In practice, these values must also be sent to the verifier from the prover, so the prover must also "commit" to these values via the hash. Instead, in Construction 13.12, $r := H(I, m)$, where $I = g^k$ for random k and message m. Since this does not include y in the hash, show how a malicious prover could choose y to forge a signature. Can you think of scenarios

where this attack would not be a problem, and scenarios where it would be a problem?

6. This question describes a bug in digital signatures present in Java from 2020 until April this year.

   (a) In the last step of the DSA algorithm, the verifier must compute $g^{\alpha s^{-1}}$. Show that if q (a prime) is the order of g, that if we define $x := s^{q-2} \mod q$, then $g^{s^{-1}} = g^x$.

   (b) If we compute s^{-1} using the method of the last question, what happens when we compute the inverse of 0?

   (c) In the non-interactive version of DSA (Construction 13.13), suppose the function F has the property that $F(1) = 0$. Show that if we use the previous method to compute inverses, and we forget to check that $r, s \neq 0$, that the signature $(0, 0)$ will pass the verification for any message and any public key.

   It might seem strange to use a hash function F such that $F(1) = 0$, but in fact the elliptic curve DSA does precisely this!

3.2 Group project

Digital signatures are a simple form of a zero-knowledge proof. In this project you'll prove the security of a slightly more complicated zero-knowledge proof: The Chaum-Pederson proof.

1. Consider the discussion in section 13.5.2, arguing that passive eavesdropping cannot help an attacker. We refer to this as "honest verifier zero-knowledge". Create a formal definition of honest-verifier zero-knowledge, ensuring that the Schnorr identification scheme satisfies this definition.

2. Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two groups with the same prime order q. Let g_1 and g_2 be two generators. Alice has one private key $x \in \mathbb{Z}_q^*$ two public keys g_1^x and g_2^x (the same exponent for each). She wants to prove to Bob that the exponents are the same, so she and Bob engage in the following protocol:

Alice (Prover)		Bob (Verifier)
$r \leftarrow \mathbb{Z}_q^*$		
$t_1 \leftarrow g_1^r$		
$t_2 \leftarrow g_2^r$	$\xrightarrow{t_1, t_2}$	
		$c \leftarrow \mathbb{Z}_q$
$s \leftarrow r + cx \mod q$	$\xrightarrow{s}$	

   Construct a method for Bob to verify this.

3. Show that this scheme is honest-verifier zero-knowledge, according to your definition.

4. Show that this scheme is secure in the sense of Definition 13.8 (you'll need to define what the public and secret key should be).

5. Construct a non-interactive version of this scheme, i.e., a single message that Alice can send to Bob that proves the same equality.

6. In certain applications, we want to anonymise a public/private keypair (g, g^x) to (h, h^x). This can be easily done by picking a random r and setting $h := g^r$, and $h^x = (g^x)^r$. Suppose Alice randomizes her key in this way and want to prove to Bob that the new key matches the old one. They could use the protocol just described, but with $\mathbb{G}_2 = \mathbb{G}_2$.

   (a) Will that affect your security proofs?

   (b) Here Alice is allowed random values in the second keypair. Is your non-interactive scheme vulnerable to the attack of problem 5? If so, construct a scheme that is not vulnerable. If your scheme was already secure, construct a scheme that *is* vulnerable.

4 Factoring algorithms

4.1 Individual guided reading

Early Factoring Algorithms. The main references for this section are:

- Chapter 10.1 and 10.3 from Katz-Lindell. 10.3 can also be helpful to understand the quadratic sieve.
- This survey by Pomerance, `https://math.dartmouth.edu/~carlp/PDF/qs08.pdf`, especially the discussion of the *sieving* component of the quadratic sieve.

For a more advanced treatment of the same topics, Antoine Joux's book [5] covers this material in Chapters 4.2 and 15.

1. Explain the difference between a general purpose and a special purpose factoring algorithm. Give a special-purpose factoring algorithm that can find a prime factor of a number in polynomial time, which will succeed for 99% of randomly chosen numbers.
2. Give all values of α and c such that $L_N[\alpha, c]$ is polynomial in $\lg N$. Give all values of α and c such that $L_N[\alpha, c]$ is exponential in $\lg N$.
3. Define a factor base. Which steps of the quadratic sieve take longer with a larger factor base, and which steps take less time?
4. Consider the matrix B of exponents produced during relation generation of the quadratic sieve (to factor a number N).
   (a) Why do we consider this matrix modulo 2?
   (b) What is the maximum weight of any row in this matrix?
   (c) What are the dimension of this matrix?
   (d) Suppose B is an $n \times m$ matrix for some n and m, describe a data structure that stores the matrix with only $O(n \cdot \log^k N)$ memory, for some k.
5. Provide pseudo-code (or real code!) for the sieving step of the quadratic sieve. Show that to test K numbers requires $O(K \log^{k_1} N)$ memory and time $O(K \log\log B \cdot \log^{k_2} N)$ (where k_1 and k_2 are arbitrary integers, and B is the smoothness bound).

The Number Field Sieve. The following two references are the recommended reading to explain the number field sieve:

- The survey by Pomerance presents a good overview of different factoring algorithms, though without any technical detail:`http://www.ams.org/notices/199612/pomerance.pdf`
- This paper presents all the technical details: `https://www.math.leidenuniv.nl/~psh/ANTproc/04psh.pdf`

If points remain unclear from these papers, this thesis can be helpful to clear up some details: `https://personal.math.vt.edu/brown/doc/briggs_gnfs_thesis.pdf`.

The original paper on the number field sieve also covers the algorithm in detail, but is probably less approachable: `https://scholarlypublications.universiteitleiden.nl/handle/1887/2149`

1. Explain why the random samples $(a + bm, N(a + b\alpha))$ can be chosen so that they are more likely to be smooth than the samples in the quadratic sieve (Hint: $N(a + b\alpha) = (-b)^d f(a/b)$).

2. Assuming the $o(1)$ part of $L[\alpha, c]$ is 0, compute the complexity of trying to factor a 2048-bit integer with the quadratic sieve ($\alpha = \frac{1}{2}$, $c = 1$) and with the number field sieve ($\alpha = \frac{1}{3}$, $c = 1.923$).

3. Suppose next year someone invents a factoring algorithm with $\alpha = \frac{1}{3}$ and $c = 0.923$. What is the new hardness of factoring 2048-bit numbers? How large do numbers need to be to reach the security that 2048-bit numbers used to have?

   Repeat the same exercise, but assume a new factoring algorithm with $\alpha = \frac{1}{4}$ and $c = 2$.

4. If you are given x and y (with $x \not\equiv \pm y \mod N$) such that $x^k \equiv y^k \mod N$ for an integer $k \geq 2$, could you recover a non-trivial factor of N?

5. Explain the issues with finding squares and square roots in the number field sieve compared to finding squares and square roots in the quadratic sieve.

Number Theory Exercises. The number field sieve relies on a great deal of number theory. This may be tricky if you do not have number theory background, though it can be difficult either way, since much of the number field sieve works over $\mathbb{Z}[\alpha]$, *not* the ring of integers, so many basic results in number theory must be reconsidered. These exercises should guide you through most of the basic results necessary.

Let $\mathbb{Z}[\alpha] = \mathbb{Z}[x]/f(x)$ for an irreducible monic polynomial f of degree d (so that $f(\alpha) = 0$ in $\mathbb{Z}[\alpha]$).

1. Show that $\mathbb{Z}[\alpha]/\mathfrak{p}$ is a finite field for any prime ideal $\mathfrak{p}$.

   (From this, it follows that all prime ideals in $\mathbb{Z}[\alpha]$ are maximal).

2. From the above, show that all prime ideals $\mathfrak{p}$ contains the principal ideal (p) generated by a prime integer p, and that this prime is unique (we say that $\mathfrak{p}$ lies over p). From this, the *degree* of $\mathfrak{p}$, defined as $[\mathbb{Z}[\alpha]/\mathfrak{p} : \mathbb{Z}/p\mathbb{Z}]$, is well-defined.

3. Suppose $\mathfrak{p}$ is degree d (the degree of $\mathbb{Z}(\alpha)$) and contains the prime p. Show that $\mathfrak{p}$ is principal, and find a generator.

4. Let $\phi : (\mathbb{Z}[\alpha]/\mathfrak{p}) \to \mathbb{Z}_{p^t}$ be a ring homomorphism. Show that if $\phi(\alpha) \in \mathbb{Z}_p$, then $\mathfrak{p}$ must have degree 1.

5. Show that the degree-1 prime ideals are in one-to-one correspondence with pairs (p, r) where $p \in \mathbb{Z}$ is prime and $r \in \mathbb{Z}_p$ is such that $f(r) \equiv 0 \mod p$.

6. Let $a + b\alpha \in \mathfrak{p}$ with a, b co-prime. Show that $\mathfrak{p}$ is degree-1 and that $a + br \equiv 0 \mod p$ for the pair (p, r) corresponding to $\mathfrak{p}$.

7. Let $a + b\alpha \in \mathfrak{p}_1$ with a, b co-prime. Show that if $a + b\alpha \in \mathfrak{p}_2$, then $\mathfrak{p}_1$ lies over a different prime than $\mathfrak{p}_2$.

8. We define a norm on ideals by $N(\mathfrak{a}) = [\mathbb{Z}[\alpha] : \mathfrak{a}]$. Show that:

   (a) If $\mathfrak{a} \subseteq \mathfrak{b}$, then $N(\mathfrak{b})|N(\mathfrak{a})$.

   (b) If $\mathfrak{p}$ is a first-degree prime ideal over a prime p, then $N(\mathfrak{p}) = p$.

   (c) If $\mathfrak{p}$ is a first-degree prime ideal over a prime p, then $p^k|N(\mathfrak{p}^k)$.

   We state without proof that for all $\beta \in \mathbb{Z}[\alpha]$, the principal ideal generated by β, denoted (β), satisfies $N((\beta)) = |N(\beta)|$ (the first is the ideal norm, the second is the field norm).

9. The ring of integers of a number field $\mathbb{Q}(\alpha)$ is a ring $\mathcal{O}$ with $\mathbb{Z}[\alpha] \subseteq \mathcal{O} \subseteq \mathbb{Q}(\alpha)$. We won't precisely define it; for our purposes, the relevant fact is that any ideal in $\mathcal{O}$ factors uniquely into a product of powers of prime ideals.

   (a) Show that if $\mathfrak{p}$ is a prime ideal in $\mathcal{O}$, that $\mathfrak{p} \cap \mathbb{Z}[\alpha]$ is a prime ideal of $\mathbb{Z}[\alpha]$.

   (b) From the above: conclude that any prime ideal in $\mathcal{O}$ contains (p) for a unique prime in $\mathbb{Z}$.

   (c) The norm on ideals in $\mathcal{O}$ is defined in the same way as the norm over $\mathbb{Z}[\alpha]$. THat is, $N(\mathfrak{a}) := [\mathcal{O} : \mathfrak{a}]$. How many properties that you proved previously will still hold over $\mathcal{O}$?

   (d) Let p be a prime integer. Considering that (p) factors into prime ideals, show that for each prime ideal $\mathfrak{p}$ lying over p, $N(\mathfrak{p}) = p^e$ for some e.

   (e) Show that if $p|N(\beta)$ for $\beta \in \mathbb{Z}[\alpha]$, that $\beta \in \mathfrak{p}$ for a prime ideal in $\mathbb{Z}[\alpha]$ lying over p.

10. Let a and b be co-prime, and let (p, r) be a pair corresponding to a degree-1 prime ideal. Define $e_{p,r}(\gamma)$ as:

    - 0 if $a + br \not\equiv 0 \mod p$
    - The maximum integer k such that $p^k|N(\gamma)$, otherwise

    Show that for co-prime a, b

    $$N(a + b\alpha) = \pm \prod_{(p,r)} p^{e_{p,r}(a+b\alpha)} \tag{1}$$

    where the product is over primes p and $r \in \mathbb{Z}_p$ such that $f(r) \equiv 0 \mod p$.

    We state without proof that $e_{p,r}$ can be extended to all non-zero elements of $\mathbb{Z}[\alpha]$, such that $e_{p,r}(\gamma\beta) = e_{p,r}(\gamma) + e_{p,r}(\beta)$.

11. Let S be a set of co-prime pairs (a, b). Suppose that

$$\prod_{(a,b)\in S} (a + b\alpha) = \gamma^2$$

for some $\gamma \in \mathbb{Z}[\alpha]$. Show that for all (p, r),

$$e_{p,r}(a + b\alpha) \equiv 0 \mod 2.$$

Explain how this is analogous to a similar criterion for the factor base in the quadratic sieve.

12. Let $\mathfrak{p}$ be a degree-1 prime ideal corresponding to (p, r). Find a ring homomorphism $\phi_\mathfrak{p} : \mathbb{Z}[\alpha] \to \mathbb{Z}_p$ such that $\ker(\phi) = \mathfrak{p}$.

    (a) Define a group homomorphism $\chi_\mathfrak{p} : (\mathbb{Z}[\alpha]/\mathfrak{p}) \to \mathbb{Z}_2$ by composing $\phi_\mathfrak{p}$ with the Legendre symbol, and mapping $-1 \mapsto 1$ and $1 \mapsto 0$. Show that if $\beta \in \mathbb{Z}[\alpha]$ can be expressed as $\beta = \gamma^2$ for $\gamma \in \mathbb{Z}[\alpha]$, that $\chi_\mathfrak{p}(\beta) = 0$.

13. Let $\mathfrak{B}$ be a set of prime ideals in $\mathbb{Z}[\alpha]$.

    (a) What set B of primes in $\mathbb{Z}$ naturally corresponds to $\mathfrak{B}$?

    (b) Show that if β is in a product of prime ideals from $\mathfrak{B}$, that $|N(\beta)|$ is a product of primes from B.

    (c) Explain why the converse does not hold: if $|N(\beta)|$ is a product of primes from B, it does not hold that β is in a product of ideals from $\mathfrak{B}$. What extra condition do we need to add to $\mathfrak{B}$ and/or β to ensure the converse holds?

14. It turns out one can define $e_\mathfrak{p} : \mathbb{Z}[\alpha] \to \mathbb{Z}$ such that, for all $\beta \in \mathbb{Z}[\alpha]$,

$$|N(\beta)| = \prod_\mathfrak{p} [\mathbb{Z}[\alpha] : \mathfrak{p}]^{e_\mathfrak{p}(\beta)}.$$

(that is, not just for first-degree number fields). Let $V \subseteq \mathbb{Z}[\alpha]$ contain all β such that $e_\mathfrak{p}(\beta)$ is even for all β. Show that V contains the

$$\{\beta : \beta = \gamma^2, \gamma \in \mathbb{Z}[\alpha]\}$$

15. Explain why, in general, if $\beta \in V$, there is not necessarily any $\gamma \in \mathbb{Q}(\alpha)$ such that $\gamma^2 = \beta$.

16. Explain why, in general, if $x^2 \in \mathbb{Z}[\alpha]$ for some $x \in \mathbb{Q}(\alpha)$, that does not imply that $x \in \mathbb{Z}[\alpha]$.

4.2 Group project

The goal of this group project is to become familiar with the details of the number field sieve and the use of heuristics in cryptography.

1. Write out all the steps of the number field and analyse the run-time of each step (in terms of whatever parameters of the algorithm you need).

2. Find all the points where you need heuristic assumptions to make statements about the run-time.

3. Formulate precise conjectures for each heuristic, such that if the conjecture were proven true, then the run-time you claim would hold.

4. Comment on the use of heuristics in cryptography and especially cryptanalysis. In what scenarios should you rely on heuristic assumptions, and in what scenarios should you not rely on such assumptions?

5 Breaking RSA and variants without the number field sieve

The main reference for this part are Dan Boneh's paper "20 years of attacks on RSA cryptosystem" [1].

5.1 Individual guided reading

1. Check that recovering the decryption key of RSA cryptosystem is computationally equivalent to factoring.
2. Quickly scan through the paper *Ron was wrong, Whit is right* by Lenstra et al. What are the main conclusions?
3. One may want to accelerate RSA decryption by using small decryption exponents instead of random ones. Study Wiener's attack in this context. How is this attack not solving the general factoring problem?
4. Solving an equation of the form $f(x) = 0 \bmod N$ where f is an integer coefficient polynomial and the factorization of N is unknown is believed to be a hard problem in general. Prove that it is computationally equivalent to factoring in the case $f(x) = x^2$.
5. Read Theorem 3.
   - How strong is the requirement on X? Use a counting argument to evaluate the expected size of the smallest solution of a randomly chosen polynomial f modulo N.
   - What is the cost of the algorithm?
6. Give conditions for a root of f modulo N to also be a root of f over the integers.
7. What is a lattice?
8. How large can the smallest vector be in a lattice? Understand the so-called Minkowski's bound.
9. Computing the shortest vector in a lattice is a NP-hard problem, but computing *rather short* problems is *easier*. In particular, what are the guarantees offered by the LLL algorithm?
10. In the lattice definition p7, why are the polynomials $g_{d,v}$ not included?
11. Can you generalize Theorem 3 to the case of two variables? Study the sketch provided in Antoine Joux's book on *Algorithmic cryptanalysis* [5], Section 13.2.2.
12. How can a bivariate generalization of Coppersmith's theorem be useful to factor when the decryption exponent is small?
13. Study Hastad's broadcast attack. Is the attack model realistic? To what extent does it/not constitute a factoring attack?

14. Study Franklin-Reiter's related message attack described in Section 4.3. Is the attack model realistic? To what extent does it/not constitute a factoring attack?

15. Study Coppersmith's short padding attack described in Section 4.4. Is the attack model realistic? To what extent does it/not constitute a factoring attack?

16. Study the partial key exposure attack described in Section 4.5. Is the attack model realistic? To what extent does it/not constitute a factoring attack?

17. Explain how measuring the power consumption of a smart card when computing an RSA signature or decryption, can lead to a full key recovery ?

18. Understand the so-called *Bellcore attack* [2]; what extra power is provided to the attacker here? To what extent does it/not constitute a factoring attack?

5.2 Group project

The goal of this group project is to get some hand-on experience on Coppersmith's techniques and their various applications to RSA.

1. Summarize Boneh-Durfee's attack in the paper *New Results on the Cryptanalysis of Low Exponent RSA* [3]. Implement the attack and describe your results

2. Using Google, identify and summarize further progress on this problem and related ones using Coppersmith's techniques.

References

[1] Dan Boneh. Twenty years of attacks on the rsa cryptosystem. *NOTICES OF THE AMS*, 46:203–213, 1999.

[2] Dan Boneh, Richard A. DeMillo, and Richard J. Lipton. On the importance of eliminating errors in cryptographic computations. *J. Cryptol.*, 14(2):101–119, 2001.

[3] Dan Boneh and Glenn Durfee. Cryptanalysis of RSA with private key d less than $n^{0.292}$. *IEEE Trans. Inf. Theory*, 46(4):1339–1349, 2000.

[4] Steven D. Galbraith. *Mathematics of Public Key Cryptography*. Cambridge University Press, 2012.

[5] Antoine Joux. *Algorithmic Cryptanalysis*. Chapman Hall/CRC, 1st edition, 2009.

[6] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography*. Chapman and Hall/CRC Press, 2007.