

RECIPROCITY LAWS
AND IDENTITY-BASED ENCRYPTION

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF MATHEMATICS
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Daniel Nelson Dore
March 2023

© 2023 by Daniel Nelson Dore. All Rights Reserved.

Re-distributed by Stanford University under license with the author.



This work is licensed under a Creative Commons Attribution-Noncommercial-Share Alike 3.0 United States License.

<http://creativecommons.org/licenses/by-nc-sa/3.0/us/>

This dissertation is online at: <https://purl.stanford.edu/vp300vn1983>

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Brian Conrad, Primary Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Ravi Vakil

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Jan Vondrak

Approved for the Stanford University Committee on Graduate Studies.

Stacey F. Bent, Vice Provost for Graduate Education

This signature page was generated electronically upon submission of this dissertation in electronic format.


Dedication

To Extra and Phantom, truly the best friends a guy could ask for!

Acknowledgments

My PhD experience has been a long and winding road, full of unexpected turns. I certainly could not have made it to the end without the generosity, love, and support of my community.

I'd like to thank my advisor Brian Conrad for his help preparing this thesis: I've learned quite a lot from him over the years of my PhD, one of the more important of which being the art of careful, precise, and detailed technical communication (and the vital role of Hagoromo chalk therein!). I'd like to thank Dan Boneh, who suggested this problem along with many others, who patiently helped me make the leap into cryptography, and whose research work has been vital to the development of the subject of this thesis.

The Stanford math department has been a welcoming and supportive academic and social home over the past seven years, and the importance of the relationships I've made there goes far beyond the world of mathematics. Among the many people in the department who have had an important impact on my life, my thesis readers Jan and Ravi have been especially significant mentors. I'd like to thank Jan for his patient, supportive, and flexible role helping me navigate the many 's in my academic career. I'd like to thank Ravi for his kindness, his fearless enthusiasm and contagious curiosity, and his tireless and sincere commitment to inclusion and community in the math department: many of my fondest Stanford memories were formed among the algebraic geometry network. And of course, I'd like to thank Gretchen for caring about all of the grad students on a personal level, being a consistent friendly face in the halls, and helping me out of my many administrative headaches.

It's said that the real PhD is the friends you made along the way, right? I can't possibly name all of the wonderful humans with whom I've shared this wild journey, but I can't help but try to name a few: To Yuval, Vivian, and Andrea: thank you for filling the Hippopede with banter and books, spaghetti and splüj. To Erin, Megan, and Jacob: I couldn't have asked for better company to watch the world burn. To Dani: *until we win!* To isis, whom I suspect is a literal wizard, and whose support and mentorship in the past year has certainly had a magical effect on me. To Psi: for introducing me to the wonderful world of cryptography as well as leopard sharks and shrimps.

Above all, to my family: Laura, Cara, Graham, Brad, Mom, and Dad. You've always believed in me, even when I did not believe in myself. There is simply no way I could have made it to this point without each of you in my life. I love you.

Contents

Dedication	iv
Acknowledgments	v
1 Introduction	1
1.1 Quadratic Residues and the Jacobi Symbol	1
1.2 Computing the Jacobi Symbol and the Quadratic Residuosity Problem	5
1.3 Cocks' IBE scheme	6
1.4 Extending Cocks' IBE	11
2 Preliminaries	14
2.1 Notation and conventions	14
2.2 Statistics, information, and security	16
2.3 Identity-based encryption	18
2.4 RSA Keys	19
2.5 Definitions of security notions	19
3 The abstract IBE construction	22
3.1 Algebraic setup	22
3.2 Defining the symbol	24
3.3 The master key	26
3.4 The isogeny image assumption	28
3.5 Extracting private keys	30
3.6 Encryption	31
3.7 Decryption	34
3.8 Security	35
4 Cocks-K�ummer IBE for number fields	45
4.1 The ring of K -integers mod N	45

4.2	The Kummer Sequence	49
4.3	Computing the symbol by factoring	49
4.4	Kummer theory in terms of Galois groups	50
4.5	The Hilbert symbol	51
4.6	The Artin reciprocity law	53
4.7	Computing the power residue symbol	55
4.8	Coda: Recovering the BLS construction	56
Bibliography		57

Chapter 1

Introduction

The problem of constructing identity-based encryption (IBE) is a venerable challenge in cryptography, originally posed by Shamir [23] in 1984. The challenge is as follows: we want Alice to be able to send a secret message to Bob without first having to obtain Bob's public key. In place of the public key, Alice should only need to know some arbitrary identifier I_{Bob} for Bob (e.g. Bob's email address). For this to be possible, we have a central trusted party - Trudy - who possesses a *master key* allowing her to give Bob a secret key which can be used to decrypt messages encrypted to I_{Bob} . One major advantage of IBE over ordinary public-key encryption is that it saves a round of communication between Alice and Bob: Alice does not need to receive any information in order to encrypt her message. This can be useful if communication costs are a bottleneck for the desired application. Moreover, IBE can be used as a cryptographic primitive inside of other protocols.

The first IBE scheme was constructed by Boneh-Franklin in 2001 [2] using the Weil pairing on (the group of \mathbf{F}_p -rational points of) elliptic curves. Since then, many other pairing-based constructions have appeared: see e.g. [9] for an overview.

The problem of constructing IBE schemes that do not rely on elliptic curve pairings has received comparatively little attention. The first such construction is an elegant scheme introduced by Cocks in [11] in 2001, based on the arithmetic of quadratic residues. Before we describe this further, we recall the key facts from the theory of quadratic residues. A classic reference for this subject is [15].

1.1 Quadratic Residues and the Jacobi Symbol

Definition 1.1.1 (Quadratic residues). Given an integer $N \geq 1$, an integer a which is coprime to N is said to be a *quadratic residue* modulo N if it is a square in the ring $\mathbf{Z}/N\mathbf{Z}$; that is, if there is some integer b such that $a \equiv b^2 \pmod{N}$.

The main feature of quadratic residues which Cocks' scheme exploits is that determining whether

a given integer is a quadratic residue modulo a *prime* may be done efficiently, but if the modulus is a composite (and not a prime power), the question of determining whether an integer is a quadratic residue is believed to be computationally intractable, with all known methods requiring factoring the modulus.

Recall that the Chinese Remainder Theorem tells us that if N, M are coprime integers greater than 1, then we have an isomorphism of rings

$$\mathbf{Z}/(NM)\mathbf{Z} \xrightarrow{\sim} \mathbf{Z}/N\mathbf{Z} \times \mathbf{Z}/M\mathbf{Z}$$

Thus, an integer a is a quadratic residue modulo the composite NM if and only if it is a quadratic residue modulo both N and M . Moreover, as a consequence of Hensel's lemma, an integer a is a quadratic residue modulo an *odd* prime power p^k if and only if it is a quadratic residue modulo p .¹ Therefore, for an arbitrary odd integer $N > 1$, *given the prime factorization of N* , we can reduce the problem of determining quadratic residues modulo N to the problem of determining quadratic residues modulo the prime factors of N .

Now, let p be an odd prime number. The ring $\mathbf{Z}/p\mathbf{Z}$ is a *finite field* \mathbf{F}_p , and as such its unit group is *cyclic* of order $p-1$. As p is odd, the squares in $(\mathbf{F}_p)^\times$ form the unique index-two subgroup, which is also the kernel of the map $x \mapsto x^{\frac{p-1}{2}}$. Therefore, we may detect quadratic residues modulo an (odd) prime p using the *Legendre symbol*:

Definition 1.1.2 (Legendre Symbol). For an odd prime p and an integer a coprime to p , the *Legendre symbol* is:

$$\left(\frac{a}{p}\right) \equiv a^{\frac{p-1}{2}} \pmod{p} = \begin{cases} 1 & a \text{ is a square modulo } p \\ -1 & a \text{ is not a square modulo } p \end{cases}.$$

We note the following properties:

Proposition 1.1.3. *Suppose p is an odd prime number.*

- *The Legendre symbol is a homomorphism from $(\mathbf{Z}/p\mathbf{Z})^\times$ to $\{\pm 1\}$, that is:*

$$\left(\frac{ab}{p}\right) = \left(\frac{a}{p}\right) \left(\frac{b}{p}\right) \tag{1.1.1}$$

- *We have*

$$\left(\frac{-1}{p}\right) = (-1)^{\frac{p-1}{2}} = \begin{cases} 1 & p \equiv 1 \pmod{4} \\ -1 & p \equiv 3 \pmod{4} \end{cases}$$

¹For the curious: it turns out that an odd integer is a quadratic residue modulo 2^k for $k \geq 3$ if and only if it is a quadratic residue modulo 8, i.e. if and only if it is congruent to 1 modulo 8

- We have

$$\left(\frac{2}{p}\right) = (-1)^{\frac{p^2-1}{8}} = \begin{cases} 1 & p \equiv 1, 7 \pmod{8} \\ -1 & p \equiv 3, 5 \pmod{8} \end{cases} \quad (1.1.2)$$

Observe that the definition of the Legendre symbol above allows us to immediately deduce an efficient² algorithm to compute it: “double and add”. This works for any group: to compute a^m , write $m = \sum_{i=1}^k \epsilon_i 2^i$ with $\epsilon_i \in \{0, 1\}$, so $a^m = \prod_{i=1}^k (a^{\epsilon_i})^{2^i}$. Each term of this product can be computed with $i \leq k$ squaring operations, and $k \approx \log m$. So as long as the group multiplication is computable in time polynomial in $\log m$, computing m -th powers is as well.

Thus, we have proved:

Theorem 1.1.4. *If N is an odd integer whose prime factorization is known and $a \in (\mathbf{Z}/N\mathbf{Z})^\times$, there is an algorithm to determine whether a is a quadratic residue modulo N whose runtime is bounded by a polynomial in $\log N$.*

This is the first ingredient needed for Cocks’ IBE scheme: the factorization of N will play the role of the “master secret key”, and integers modulo N will play the role of “public keys”. Actually, in order to be able to derive the secret key for a given public key, we need the following stronger result:

Theorem 1.1.5. *If N is an odd integer whose prime factorization is known and a is a quadratic residue modulo N , then there is an algorithm whose runtime is bounded by a polynomial in $\log N$ which computes a “square root” of $a \pmod{N}$.*

Note that the Chinese Remainder Theorem implies that number of square roots of a modulo N is equal to 2^k where k is the number of prime factors of N : for each prime factor p of N , we may multiply a given square root by an integer which is congruent to -1 modulo $p^{\text{ord}_p(N)}$ and to 1 modulo $q^{\text{ord}_q(N)}$ for every other prime factor of N .

Proof. As the prime factorization of N is known, it suffices to know a square root of a modulo p for each prime factor p of N . Indeed, suppose we are given that $N = p_1^{e_1} \cdots p_k^{e_k}$. If we know square roots of a modulo each p_i , such square roots may be lifted from $\mathbf{Z}/p_i\mathbf{Z}$ to $\mathbf{Z}/p_i^{e_i}\mathbf{Z}$ by Hensel’s lemma:³ it only requires a single modular inversion to lift a solution from $\mathbf{Z}/p_i'\mathbf{Z}$ to $\mathbf{Z}/p_i^{\nu+1}\mathbf{Z}$, and this happens $e_i = O(\log N)$ times. Once we have square roots modulo each $p_i^{e_i}$, we can use (the extended version of) the Euclidean algorithm to combine them via the Chinese Remainder Theorem.

Thus, it suffices to consider the case where N is a prime number p , so we are computing square roots in the finite field \mathbf{F}_p . Efficient algorithms to do this go all the back to work of Tonelli from the nineteenth century! See e.g. [12, §1.5.1] for a modern treatment.

□

²Here and always, “efficient” means “with runtime bounded by a polynomial in $\log N$ ”. It is useful to think of N as a random string of $\log N$ bits. We’ll define these notions more carefully in Chapter 2.

³NB: this is where we use the assumption that N is odd

The security of Cocks' scheme is based on the assumption that a partial converse is true: it should be computationally intractable to determine whether certain elements of $(\mathbf{Z}/N\mathbf{Z})^\times$ are quadratic residues without knowing the factorization of N . In order to state this precisely, we use the following extension of the notion of a Legendre symbol:

Definition 1.1.6 (Jacobi symbol). For an odd *positive* integer M with prime factorization $M = p_1^{e_1} \cdots p_k^{e_k}$ and $a \in (\mathbf{Z}/M\mathbf{Z})^\times$, the *Jacobi symbol* is defined as:

$$\left(\frac{a}{M}\right) := \left(\frac{a}{p_1}\right)^{e_1} \cdot \left(\frac{a}{p_2}\right)^{e_2} \cdots \left(\frac{a}{p_k}\right)^{e_k}$$

In other words, the Jacobi symbol $\left(\frac{a}{M}\right)$, as a function of the two odd integers a, M with $M > 0$, is the *unique* extension of the Legendre symbol which is a homomorphism in both a and M , i.e. such that the following holds, analogously to (1.1.1):

$$\left(\frac{a}{bc}\right) = \left(\frac{a}{b}\right) \left(\frac{a}{c}\right) \quad (1.1.3)$$

We note a few other algebraic properties of the Jacobi symbol, all of which follow directly from the analogous properties of the Legendre symbol.

Proposition 1.1.7. *Let N be an odd positive integer and a an integer.*

1. *The Jacobi symbol is multiplicative in both arguments. In particular, if a is a quadratic residue modulo N , then $\left(\frac{a}{N}\right) = 1$.*

2. *The Jacobi symbol $\left(\frac{a}{N}\right)$ only depends on the integer a modulo N .*

3.

$$\left(\frac{-1}{N}\right) = (-1)^{\frac{N-1}{2}} \quad (1.1.4)$$

4.

$$\left(\frac{2}{N}\right) = (-1)^{\frac{N^2-1}{8}} = \begin{cases} 1 & N \equiv 1, 7 \pmod{8} \\ -1 & N \equiv 3, 5 \pmod{8} \end{cases} \quad (1.1.5)$$

Remarkably, the Jacobi symbol $\left(\frac{a}{N}\right)$ can be computed efficiently *without* knowing the factorization of N . This fact is crucial: as we will see below, the encryption of a message is defined in terms of a Jacobi symbol. The method to calculate the Jacobi symbol comes from an extended version⁴ of the law of quadratic reciprocity:

⁴The original version of the law of quadratic reciprocity, as proved by Gauss, is the special case of this statement when a, b are odd primes.

Theorem 1.1.8 (Quadratic Reciprocity). *Let a, b be two odd positive integers which are coprime to each other. Then the following identity holds:*

$$\left(\frac{a}{b}\right) \cdot \left(\frac{b}{a}\right) = (-1)^{\frac{a-1}{2}} \cdot (-1)^{\frac{b-1}{2}} = \begin{cases} 1 & a \equiv b \pmod{4} \\ -1 & a \equiv -b \pmod{4} \end{cases} \quad (1.1.6)$$

This is one of the most celebrated theorems in the history of number theory: the observation of the pattern dates back to Euler and Lagrange, and the law was first proved by Gauss, who upon discovering it proceeded to prove it seven more times!

1.2 Computing the Jacobi Symbol and the Quadratic Residuosity Problem

For our purposes, the value of the law of quadratic reciprocity comes from the following application:

Theorem 1.2.1. *For an odd positive integer N and an integer a which is coprime to N , there is an algorithm to compute the Jacobi symbol $\left(\frac{a}{N}\right)$ with runtime $O(\log a * \log N)$.*

Proof. The technique will resemble the Euclidean division algorithm used to compute the greatest common divisor of two numbers. It will proceed by repeatedly using quadratic reciprocity to “flip” the Jacobi symbol to be computed, then reducing the “numerator” modulo the “denominator”.

More precisely:

- Let $\alpha = a$, $\beta = N$, and set $t = 1$. The quantity t will be a running value which will hold the result at the end of the loop.
- While α is nonzero, repeat:
 - Write $\alpha = 2^m \cdot \alpha_0$, so $\left(\frac{\alpha}{N}\right) = \left(\frac{2}{N}\right)^m \cdot \left(\frac{\alpha_0}{N}\right)$. If N is congruent to 3 or 5 modulo 8, set $t = (-1)^m \cdot t$. In either case, replace α with α_0 .
 - Apply quadratic reciprocity: set $(\alpha, \beta) = (\beta, \alpha)$. If both α and β are congruent to 3 modulo 4 - so $(-1)^{\frac{\alpha-1}{2} \cdot \frac{\beta-1}{2}} = -1$ - replace t with $-t$.
 - Reduce α modulo β .

To analyze the runtime of this program, note that it is literally the Euclidean algorithm used to compute the greatest common divisor of two numbers, except for the fact that at each step, we do a tiny constant amount of work to check the value of $\alpha, \beta \pmod{4}$ and update the running tally t appropriately. In addition, the step in each iteration of the loop where we factor out the largest power of 2 dividing $\alpha = 2^m \alpha_0$ may be done in time proportional to $m = O(\log \alpha) = O(\log N)$,

by simply counting the number of trailing zeros in the bit representation of α and then shifting it appropriately.

The dominant cost of running this algorithm therefore consists of the repeated modular reductions at each step of the loop, precisely as in the classical Euclidean algorithm. See e.g. [18, Volume 2, §4.5.3] for a precise analysis of the runtime of the Euclidean algorithm.

Roughly, we expect that in each round, the size of $|\alpha| + |\beta|$ is reduced by a constant factor of $3/4 < 1$, so the number of iterations of the loop is logarithmic.

We can see this by breaking into cases to see what happens when we reduce α modulo β . In the first case $\beta \leq \alpha/2$, we have $\alpha' < \beta \leq \alpha/2$. In the second case $\beta > \alpha/2$, we have $\beta < \alpha < 2\beta$ and thus $\alpha' = \alpha - \beta < \alpha/2$. In either case, since $\alpha > \beta$, and hence $2\alpha > \alpha + \beta$, so $\alpha/2 > \frac{1}{4}(\alpha + \beta)$; thus, subtracting $\alpha/2$ from α reduces the sum by at least $1/4$ of its size.

Moreover, computing modular reduction of α modulo β by repeated subtraction has runtime proportional to $\log \beta * (\log \alpha - \log \beta)$. The resulting sum telescopes, i.e. the above analysis lets us compute:

$$\text{Time}(a, N) \leq \sum_{i=1}^{\log N} \log(\alpha) * (\log((3/4)^i \alpha) - \log((3/4)^{i+1} \alpha)) \quad (1.2.1)$$

$$\leq \log(\alpha) * \sum_{i=1}^{\log N} \log(4/3) \quad (1.2.2)$$

$$= O(\log N * \log \alpha) \quad (1.2.3)$$

□

Due to this result, we see that the Jacobi symbol provides a computable “obstruction” to an integer being a square modulo N : one can compute the Jacobi symbol $(\frac{a}{N})$, and if it is -1 conclude that a cannot be a square modulo N . However, assuming that $(\frac{a}{N}) = 1$, for N with at least two prime factors, no known algorithm can efficiently determine whether a is a quadratic residue modulo N . This is known in the cryptography literature as the Quadratic Residuosity (QR) problem:

Problem 1.2.2 (Quadratic Residuosity Problem). Given a composite integer N which is not a prime power, and some $a \in (\mathbf{Z}/N\mathbf{Z})^\times$ such that the Jacobi symbol $(\frac{a}{N})$ is equal to 1, determine whether a is a quadratic residue: i.e. whether there exists some $b \in (\mathbf{Z}/N\mathbf{Z})^\times$ with $a = b^2$.

1.3 Cocks’ IBE scheme

With the above arithmetic preliminaries out of the way, we are ready to discuss Cocks’ construction of identity-based encryption. First, we have the construction of the “master key”, used to derive secret keys for any identity.

Definition 1.3.1. Choose some integer λ as the key length. To initialize the IBE scheme, we first choose two random odd primes p, q such that $N = pq$ has size approximately 2^λ . The *master public key* is N , and the *master secret key* consists of the factorization $N = pq$.

For those familiar with the RSA cryptosystem, this is exactly the same as an RSA keypair. The security of the system is based on the problem of factoring integers, which is believed to be intractable for e.g. $\lambda \geq 2048$.

By the above discussion, knowledge of the secret key is sufficient not only to determine whether some $a \in (\mathbf{Z}/N\mathbf{Z})^\times$ is a quadratic residue, but also to find a square root in the case that it is a quadratic residue, per Theorem 1.1.5. This will be the basis for the allocation of secret keys.

In ordinary public-key cryptography, the secret key is usually chosen at random, and the public key is the result of evaluating some one-way function on the secret key. For example: the secret key could be a pair of large primes and the public key their product, as in the RSA system, or the secret key could be an integer modulo the order of a finite commutative group with a hard discrete logarithm problem, and the public key the result of multiplying a base point by this number. This means that it is impossible to predict the public key or to choose it in advance, without knowledge of the secret key.

This is where the challenge of IBE comes from: we want to be able to assign “public keys” to users in a public, predictable, deterministic manner. At the same time, it should be impossible to compute the corresponding secret keys without knowledge of the master secret. Therefore, IBE requires a “universal trapdoor function”: the function mapping secret keys to public keys should be a cryptographic “one-way function” (i.e. impossible to invert in polynomial time), but the “master secret” should be sufficient to invert this function for *any* secret key.

By the above, we see that quadratic residues and square roots provide the needed properties: by Theorem 1.1.5, the knowledge of the prime factorization of an odd composite number N is sufficient to determine which numbers are quadratic residues, and to compute square roots for those that are. But at the same time, when N is not a prime power, the Quadratic Residuosity Problem 1.2.2 is believed to be intractable.

Using this idea, we are led to Cocks’ method of assigning keys: roughly, we’d like to have the “public key” be a quadratic residue in $(\mathbf{Z}/N\mathbf{Z})^\times$, chosen as a deterministic function of the user identity, and the corresponding secret key its square root. However, as knowledge of the factorization of N is needed to distinguish between genuine quadratic residues and elements of $(\mathbf{Z}/N\mathbf{Z})^\times$ with Jacobi symbol 1, we need to modify this idea slightly. To make this work, we add the following information to the public parameters (i.e. the “master public key”) of the scheme:

Definition 1.3.2. The IBE protocol includes the following additional public parameters:

- A set of *identities* \mathcal{ID} .

- Let \mathcal{H} be a cryptographic hash function with domain \mathcal{ID} which is valued in the set of elements a of $(\mathbf{Z}/N\mathbf{Z})^\times$ such that $\left(\frac{a}{N}\right) = 1$.
- Let $u \in (\mathbf{Z}/N\mathbf{Z})^\times$ be an element with $\left(\frac{u}{N}\right) = 1$, but which is *not* a quadratic residue: in other words, u is such that $\left(\frac{u}{p}\right) = \left(\frac{u}{q}\right) = -1$.

Here, a *cryptographic hash function* is, roughly, a function whose output for any given choice of input is indistinguishable from a uniformly random selection from its codomain. We'll discuss this notion more rigorously in § 2.5.

To construct \mathcal{H} , we could start with any cryptographic hash function whose output is a string of bits of length $\lceil \log N \rceil$ and reduce the output mod N . Then, compute the gcd of the result with N , and if the result is not 1, start over (this happens with very low probability, roughly $O\left(\frac{1}{\sqrt{N}}\right)$: note that it implies finding a factor of N !). Then, to ensure that the result has Jacobi symbol 1, simply compute the Jacobi symbol, and if the result is -1 , start over. If the output of \mathcal{H} is modeled as a uniformly random string of bits of length $\lceil \log N \rceil$, the result is a uniformly random element of the set of elements of $(\mathbf{Z}/N\mathbf{Z})^\times$ whose Jacobi symbol is 1.

Knowing the factorization of N , it is straightforward to find an appropriate u . Alternatively, we could e.g. require in the initial setup that p, q are both congruent to 3 modulo 4. In that case, by Proposition 1.1.3, we see that taking $u = -1$ works.

The purpose of the public value u is to make it possible to find a genuine quadratic residue mod N without knowing the secret key: the publicly known hash function \mathcal{H} assigns an element v of $(\mathbf{Z}/N\mathbf{Z})^\times$ for a given user identity such that $\left(\frac{v}{N}\right) = 1$. Thus, either

$$\left(\frac{v}{p}\right) = \left(\frac{v}{q}\right) = 1,$$

and so v is a quadratic residue mod N , or

$$\left(\frac{v}{p}\right) = \left(\frac{v}{q}\right) = -1.$$

Since we know that

$$\left(\frac{u}{p}\right) = \left(\frac{u}{q}\right) = -1,$$

it follows that exactly one of $\{v, uv\}$ is a quadratic residue modulo N . Now, we can define the keypairs associated to an identity:

Definition 1.3.3. The *public key* associated to user identity $\text{id} \in \mathcal{ID}$ is defined to be $\mathcal{H}(\text{id}) \in (\mathbf{Z}/N\mathbf{Z})^\times$. The *secret key* associated to id is a choice of element $s \in (\mathbf{Z}/N\mathbf{Z})^\times$ such that either $s^2 = v$ or $s^2 = uv$.

The ambiguity of whether or not v is a quadratic residue leads to some unfortunate practical consequences: in order to encrypt a message to identity id , the sender must effectively prepare two

different ciphertexts, one corresponding to v and one to uv . This results in doubling the length of the ciphertext, with half of the ciphertext being useless. However, subsequent work on the problem has led to more space-efficient IBE schemes built from similar principles, such as [3] and [17].

Now that we have the keys, let's see how Alice can encrypt a message to Bob. A common paradigm (see e.g. [6, Chapters 10-11]) to construct “asymmetric” - i.e. public key - encryption is to start with the construction of a simpler primitive, *symmetric encryption*. This term denotes encryption based on a *shared* secret key, which is needed both for encryption and decryption. Such schemes, also referred to as “ciphers”, are frequently very efficient in both time and space requirements and have strong security properties.

The primordial cipher is the “one time pad”: to encrypt a sequence of bits m , we choose at random a key κ , which is a sequence of bits of the same length as m , and the resulting ciphertext is $m \oplus \kappa$, where \oplus is bitwise XOR (or equivalently, vector addition over \mathbf{F}_2). This scheme is perfectly secure, in that the ciphertext is a uniformly random string of bits, fully independent from m . However, it is not very practical, as it is crucial that the same key is never used more than once, and so we must securely communicate a bitstring of the same length as the original message. However, this simple construction forms the starting point for many real encryption schemes: first, a shorter key is used as a seed value to construct a *pseudo*-random string of bits, and then a message is encrypted by taking the bitwise XOR with the resulting “keystream”. An example is the widely used AES-GCM cipher suite.

In order to turn a symmetric cipher into a public-key encryption scheme, a typical method is to use a “key exchange” protocol, such as the Diffie-Hellman protocol. This consists of a method to construct a “shared secret” between two parties without the need for a secure communication channel. Alice chooses at random a secret value t , called an *ephemeral key*, and computes the shared secret encryption key κ as some function of t and Bob's public key. Then, Alice encrypts the message to Bob using κ , and transmits the result along with some “public” value computed from t , which will be enough to allow Bob to compute κ using his secret key.

In the standard Diffie-Hellman protocol, this works as follows: let \mathbf{G} be a group of order ℓ whose discrete logarithm problem is hard and g a generator of \mathbf{G} . Then Bob's secret key is an element $s \in \mathbf{Z}/\ell\mathbf{Z}$ and his corresponding public key is $P = g^s$. Alice's ephemeral key is an element $t \in \mathbf{Z}/\ell\mathbf{Z}$, and the shared secret is P^t . Then, along with the message encrypted via this secret, Alice transmits the value $T = g^t$. As $T^s = P^t$, Bob can compute the shared secret and decrypt the message. The resulting public-key encryption scheme is known as ElGamal.

Cocks' IBE scheme follows this paradigm: to encrypt a message to Bob, Alice generates a keystream κ , encrypting one bit at a time. We think of the digits as being elements of $\{\pm 1\}$, rather than $\{0, 1\}$, since we are working with multiplicative groups. *For each bit*, Alice chooses an ephemeral key $t_i \in (\mathbf{Z}/N\mathbf{Z})^\times$ uniformly at random, and sets $\kappa_i = \left(\frac{t_i}{N}\right)$. Then the encrypted bit is $c_i = \kappa_i * m_i$, with m_i the i -th bit of the message. Now, Alice must include enough additional

information in the ciphertext to allow Bob to compute κ_i using his secret key: that is, Alice must communicate the Jacobi symbol of t_i . To do this, we consider the rings $A = (\mathbf{Z}/N\mathbf{Z})[X]/(X^2 - v)$ and $\bar{A} = (\mathbf{Z}/N\mathbf{Z})[\bar{X}]/(\bar{X}^2 - uv)$, where $v = \mathcal{H}(\text{id}_{\text{Bob}})$ is Bob's public key. To disguise the value of t_i while still allowing Bob to determine $(\frac{t_i}{N})$, Alice multiplies t_i by a square in A (resp. \bar{A}), setting

$$s_i + 2X = t_i + \frac{v}{t_i} + 2X = t_i \left(1 + \frac{X}{t_i}\right)^2, \quad \bar{s}_i + 2\bar{X} = t_i + \frac{uv}{t_i} + 2\bar{X} = t_i \left(1 + \frac{\bar{X}}{t_i}\right)^2 \quad (1.3.1)$$

Then, the ciphertext is $(c_i, s_i, \bar{s}_i)_{i=1}^L$, with L the length of the message. Note that this scheme suffers from rather drastic ciphertext expansion: for each bit of the message, Alice must transmit the encrypted bit c_i along with two elements of $\mathbf{Z}/N\mathbf{Z}$, with each taking up $\lceil \log N \rceil \geq 2048$ bits of space. We'll discuss this problem a bit more in Section 1.4.

Now assume, without loss of generality, that v is a quadratic residue, so Bob's secret key s satisfies $s^2 = v$. Then, knowing s is equivalent to knowing a $\mathbf{Z}/N\mathbf{Z}$ -algebra homomorphism $A \rightarrow \mathbf{Z}/N\mathbf{Z}$: such a homomorphism is uniquely determined by mapping X to a square root s of v . Bob can apply this homomorphism to $s_i + 2X$, obtaining

$$s_i + 2s = t_i \left(1 + \frac{s}{t_i}\right)^2$$

As $1 + \frac{s}{t_i}$ is now an element of⁵ $(\mathbf{Z}/N\mathbf{Z})^\times$, we see that

$$\left(\frac{s_i + 2s}{N}\right) = \left(\frac{t_i}{N}\right) = \kappa_i$$

and thus Bob is able to recover the symmetric key, and then the message, by setting $m_i = c_i * \kappa_i$.

Now, to understand why the security of this encryption scheme reduces to the difficulty of the QR Problem 1.2.2, consider the *other* component \bar{s}_i , corresponding to the element $uv \in (\mathbf{Z}/N\mathbf{Z})^\times$. Note that as uv is not a quadratic residue but has Jacobi symbol 1, we must have $\left(\frac{uv}{p}\right) = \left(\frac{uv}{q}\right) = -1$. We argue that \bar{s}_i provides no information whatsoever about the value of $(\frac{t_i}{N})$. To see this, note that there are 4 elements τ of $(\mathbf{Z}/N\mathbf{Z})^\times$ with $\tau + \frac{uv}{\tau} = \bar{s}_i$. Indeed, these are:

$$\tau = (\tau_p, \tau_q) \in \left\{ (t_i, t_i), \left(\frac{uv}{t_i}, \frac{uv}{t_i}\right), \left(\frac{uv}{t_i}, t_i\right), \left(t_i, \frac{uv}{t_i}\right) \right\}$$

Note that these are distinct, since $uv \neq t_i^2$ modulo p and q (indeed, we are assuming uv is not a square modulo both p and q). Now, since both Legendre symbols of uv are -1 , the first two of these have $(\frac{\tau}{N}) = (\frac{t_i}{N})$, whereas the second two have $(\frac{\tau}{N}) = -(\frac{t_i}{N})$. Therefore, knowing \bar{s}_i provides no

⁵We can assume that $1 + \frac{s}{t_i}$ is a unit, which is equivalent to the assumption that $s \neq -t_i$ modulo p and q ; as t_i is drawn at random, the probability of being a non-unit is astronomically small, being $O(\frac{1}{\sqrt{N}})$. To be careful, Alice could check that the $\gcd(t_i^2 - v, N) = 1$, ensuring that this does not happen.

information as to the value of $\left(\frac{t_i}{N}\right)$.

Now, suppose an eavesdropper is able to successfully glean information about m_i from $(c_i, s_i, \overline{s_i})$. As one of the two components $(s_i, \overline{s_i})$ is statistically independent from m_i , such an eavesdropper must be able to distinguish which of the two components is relevant, i.e. which of v, uv is a quadratic residue. This is exactly Problem 1.2.2. We'll give a more detailed treatment of this security reduction later on in a much broader context, once we have precise definitions for the various cryptographic notions involved. However, here is a slightly informal statement summarizing the work we've reviewed so far (this goes back to [11, §5]):

Theorem 1.3.4 (Cocks). *The security of Cocks' IBE scheme can be reduced to the difficulty of Problem 1.2.2: if there is an efficient algorithm which can extract information about a message m from its encryption, then there is an efficient algorithm which can distinguish quadratic residues in $(\mathbf{Z}/N\mathbf{Z})^\times$ among the set of elements a with $\left(\frac{a}{N}\right) = 1$.*

1.4 Extending Cocks' IBE

Since the 2001 publication of [11], a number of works have appeared which improve upon or generalize the original method. One of the more notable is [3], which uses the arithmetic of quadratic forms to greatly reduce the ciphertext expansion, reducing the overhead from two elements of $\mathbf{Z}/N\mathbf{Z}$ per bit to one element of $\mathbf{Z}/N\mathbf{Z}$ in total plus an extra bit for each message bit.

Moving in a different direction, Boneh, LaVigne, and Sabin construct a generalization of Cocks' IBE construction which replaces quadratic residues by ℓ -th power residues for general ℓ in [4]. Unfortunately, the problem of ciphertext expansion is even worse than in the Cocks scheme, requiring up to ℓ^2 elements of $\mathbf{Z}/N\mathbf{Z}$ to encode each $\mathbf{Z}/\ell\mathbf{Z}$ -valued “digit”. However, there are some interesting new features: Clear and McGoldrick show that this scheme is additively homomorphic for messages in $\mathbf{Z}/\ell\mathbf{Z}$ in [10], i.e. that given two messages m_1, m_2 encoded as elements of $(\mathbf{Z}/\ell\mathbf{Z})^M$, the encryption of $m_1 + m_2$ can be computed from the encryptions of m_1 and m_2 . Such homomorphic properties of encryption schemes are often of great interest for the development of more sophisticated cryptographic protocols using the encryption scheme as a primitive, as they allow for trustless computation on encrypted data. For example, such homomorphic properties are useful in the realm of private cryptocurrencies such as [24], making it possible to verify that transaction amounts add up correctly without needing to reveal any information about the transaction.

The construction and analysis are very similar to the above, replacing the Jacobi symbol and the law of quadratic reciprocity with the general ℓ -th power symbol and reciprocity law. The development of these “higher reciprocity laws” formed one of the core motivating problems in algebraic number theory, extending from the late nineteenth century to today. Indeed, one of the 23 problems posed by David Hilbert in his famous centennial ICM address in 1900 asks for the most general formulation of such laws. One of the crowning achievements of twentieth century number theory was the successful

generalization of this theory, resulting in a rich description of the arithmetic of all abelian extensions of number fields (that is, extensions with abelian Galois group, such as those obtained by taking square roots). This generalization is called *class field theory*.

Inspired by [5], we develop a very general abstract algebraic framework in which to study Cocks' construction as well as the BLS generalization. As with these systems, the master secret will be the factorization of an RSA number $N = pq$. We replace the multiplicative group of units modulo N with *any* algebraic group over a finite étale extension of $\mathbf{Z}/N\mathbf{Z}$, including examples such as algebraic tori, and the operation of squaring with any isogeny among such groups. In particular, for a finite étale $\mathbf{Z}/N\mathbf{Z}$ -algebra A of constant rank and any short exact sequence of algebraic groups

$$0 \rightarrow \underline{C} \rightarrow G \xrightarrow{\varphi} H \rightarrow 0,$$

where \underline{C} is a constant cyclic group of prime order ℓ , with ℓ coprime to N , we construct an identity-based encryption scheme whose public keys are elements of $G(A)$ and whose corresponding private keys are preimages of the public keys under φ .

There are natural analogues of the Legendre and Jacobi symbols in this context, valued in \underline{C} , such that the kernel of the “Legendre symbol” is exactly the image of φ over each prime ideal of A . This leads to a natural analogue of the Quadratic Residuosity Problem 1.2.2, where we are asked to distinguish elements of $G(A)$ which are in the image of φ among the set of elements whose “Jacobi symbol” is trivial: we'll refer to this as the “isogeny image problem” for φ . Setting $G = H = \mathbf{G}_m$, φ the ℓ -th power map, then choosing a primitive ℓ -th root of unity $a \in \mathbf{Z}/N\mathbf{Z}$, and setting $A = \mathcal{O}_{\mathbf{Q}(\zeta_\ell)} / (N, \zeta_\ell - a) \simeq \mathbf{Z}/N\mathbf{Z}$, we can recover the construction of [5]. (See Section 4.8).

In this generality, we prove that the security of the corresponding IBE scheme reduces to the hardness of this “isogeny image problem”: this is Theorem 3.8.2. We need to develop a good deal of notation to give a formal statement of this result, but we can give an informal statement as follows:

Theorem 1.4.1. *Given an isogeny $\varphi: G \rightarrow H$ of algebraic groups over a finite étale $\mathbf{Z}/N\mathbf{Z}$ -algebra of constant rank, along with an identification of $\ker \varphi$ with \underline{C} for a finite cyclic group of prime order ℓ not dividing N , we construct an identity-based encryption scheme. The security of this scheme is reduced to the assumed hardness of the problem of detecting whether elements of $H(A)$ are in the image $\varphi(G(A))$.*

However, in order for the encryption function to be efficiently computable, we must restrict the generality: we need an appropriate analogue of Theorem 1.1.8, the law of quadratic reciprocity. As we saw above, this law is the key fact which enables us to compute Jacobi symbols modulo N without being able to factor N .

We obtain this analogue through class field theory: specifically, the reciprocity laws of Artin and Hilbert are far-reaching generalizations of Theorem 1.1.8. This allows us to extend the generality of [5] quite substantially, replacing $\mathbf{Z}/N\mathbf{Z}$ with $\mathcal{O}_K/N\mathcal{O}_K$ for any number field K containing a primitive

ℓ -th root of unity (and such that p, q are unramified in K). We refer to the resulting IBE scheme as “Cocks-Kummer IBE”, and develop it in detail in Chapter 4. In future work, we hope to go beyond this context, making use of modern generalizations of class field theory. For example, the case of algebraic tori should be quite feasible, and we hope that this setting might offer hope for addressing the problem of ciphertext expansion. As it stands, our construction is strictly *less* space-efficient than the constructions of Cocks (for $\ell = 2$) or BLS (for general ℓ). However, it is our hope that by demonstrating the great generality and robustness of Cocks’ construction, we are taking the first step towards a whole world of possible extensions, as well as shining light on the mathematics powering the theory.

Chapter 2

Preliminaries

2.1 Notation and conventions

We collect a few conventions and pieces of notation here for convenience. This thesis is not the place for a thorough development of the foundations of theoretical computer science, cryptography, or algebraic number theory, but we hope to be precise enough with our language to make it possible for a reader who is unfamiliar with one or more of those subjects to understand what’s going on anyway. We stress that all constructions developed here can be understood in simple, concrete terms, and that foundational concepts such as computational complexity theory or Galois cohomology are used primarily for notational efficiency.

Here is a list of some notation that will be used throughout this thesis:

- All asymptotics (e.g. “polynomial time”) are with respect to the *security parameter* λ . Intuitively, λ is supposed to represent the number of bits of security provided by a system: using the best known attacks to reduce breaking the security of the system to an equally difficult instance of brute-force guessing some bit-string, what is the length of this bit-string? Formally, λ is an omnipresent independent variable: when we initialize a system, we write e.g. “the public parameters are the result of computing $G(1^\lambda)$ ”. 1^λ refers to λ encoded in *unary* digits, resulting in λ both being the *size* and the *value* of the input.

Aside: In theoretical computer science, variables always have a size, effectively capturing the number of bits of digital memory needed to store the value of the variable. When formalizing notions in computational complexity theory, analysis of algorithms, etc., computational problems and the algorithms attempting to solve them are technically defined as an *infinite sequence* of computational objects (depending on your preferred model of computation, these could be Turing machines, formal languages, boolean circuits, λ -expressions...), one for each “size” of input, and asymptotic notions are defined in terms of taking this size to infinity. Of course, nobody ever discusses such points in papers: much like the early chapters of Bourbaki or EGA IV₁, it’s the sort of thing you glance over once, assure

yourself there's no funny business happening, and then never discuss in public again.

As such, we will frequently omit λ from the notation: but formally, it should be considered as an input variable to every function.

- $\{0, 1\}^*$ refers to the set of bit-strings of arbitrary (finite) length.
- When we discuss “algorithms” or “computational adversaries” etc., we are always referring to probabilistic polynomial time Turing machines - that is, Turing machines whose runtime is bounded by a polynomial in λ and which are allowed to “flip a coin” to make decisions. These can be modelled as algorithms that depend on an additional input variable $\mathfrak{r} \in \{0, 1\}^L$, a bit-string of some length, thought of as a “random tape”. The output of such an algorithm is then a random variable over the space of such \mathfrak{r} , sampled according to the uniform distribution.
- If not specified otherwise, if we describe choosing an element of a finite set “at random”, we mean with respect to the *uniform* distribution; if multiple elements are chosen “at random”, we mean that they are chosen *independently* from the uniform distribution. We denote such a choice with an arrow decorated with an R , for example:

$$\{x_1, x_2\} \stackrel{R}{\leftarrow} \mathcal{X}$$

refers to choosing two elements independently from the uniform distribution on a set \mathcal{X} .

- A quantity $f(\lambda)$ depending on λ (and perhaps other variables) is *negligible* if $f(\lambda) = o(\lambda^{-k})$ for all k as $\lambda \rightarrow \infty$.
- For a ring A , we denote the set of prime ideals of A as $|\mathrm{Spec}(A)|$. This set may be identified with the underlying topological space of the affine scheme $\mathrm{Spec}(A)$, but such notation is mostly just a matter of habit. While some scheme-theoretic language is used later to formulate some constructions, we make no essential use of scheme theory in this work: everything could be stated just as easily purely in terms of commutative algebra.
- An *algebraic group* over a ring A will always mean an affine group scheme which is smooth over A . When A is a field or product of fields, such group schemes may always be realized as subgroups of $\mathrm{GL}_{n,A}$ for some n , cut out by polynomial equations.

We recommend [6] as a reference on basic notions (and advanced ones) in cryptography. In particular, [6, §2.3] contains a discussion of the computational formalism we discuss above.

For the algebra and number theory side of things, we recommend [22] as a reference on Galois cohomology and local class field theory, and [20], [21] as thorough references on algebraic number theory and class field theory respectively. We do not make any serious use of any material more advanced than that which can be found in these references, though we use some language from algebraic geometry as a convenience to organize some of our later ideas.

2.2 Statistics, information, and security

To give some context for the probability calculations in this section and their relevance to cryptography, we make a small detour into Bayesian inference and information theory. These provide a way to use probability theory to assign probabilities to non-random but unknown quantities. The example relevant for cryptography is the context of attempting to use a statistical model to make inferences about an unknown message based on an observation of the encrypted ciphertext. This is the precise meaning of the “information” contained in the ciphertext. The need for Bayesian statistics instead of the more elementary “frequentist” statistics comes from the fact this is a “single-shot inference”: we don’t get to observe the result of encrypting the same message many times with different random choices. This material is not used directly in our proofs, but we felt it was helpful to orient ourselves to the importance of the statistics involved. See [6, Chapter 2] for some related discussion.

Suppose we fix some parametrized statistical model for the probability distribution of some data: a family of probability distributions $P(x; H)$, $x \in \mathcal{X}$, $H \in \mathcal{H}$, indexed by some model parameters H (or “hypothesis”), on a set \mathcal{X} . The quantity H lives in some parameter space \mathcal{H} . The value of H is unknown and cannot be observed directly, and we want to make an *inference* about the value of H from some *observations* O .

We assume that we have some *prior* distribution $P(H)$ on \mathcal{H} , reflecting our beliefs already on what the value of H is. In the absence of any initial knowledge about the data, this might be taken to be a maximum-entropy distribution, such as the uniform distribution on a finite/compact set. We are able to make observations on the data \mathcal{X} , sampling $O \in \mathcal{X}$ from some unknown real distribution $\Psi(x)$. Our goal is to find the H such that $P(\cdot; H)$ best approximates $\Psi(x)$.

Given an observation $O \in \mathcal{X}$, our model assigns a *likelihood* of this observation as a function of the hypothesis H . This is the model prediction $P(O; H)$. With this, we use Bayes’ rule to define the *posterior probability* of our hypothesis, updating our prior:

$$P_{\text{posterior}}(H; O) := \frac{P(O; H)P_{\text{prior}}(H)}{P(O)} \quad (2.2.1)$$

Here, $P(O) = \mathbf{E}[P(O; H) | H \sim P_{\text{prior}}(H)]$ is the *marginal* probability of the observation O .

In the context of cryptography, this is applied as follows: From the perspective of an outsider intercepting the communication, the message $m \in \mathcal{M}$ is an unknown and not directly observable quantity, and accordingly, our hypothesis space \mathcal{H} is the space of messages \mathcal{M} . What is observable is the ciphertext c , which is the randomized output of $\mathcal{E}(m)$ when m is the real message being encrypted. In other words, the choice of m together with the encryption algorithm $\mathcal{E}(m)$ determines a “real” probability distribution Ψ on the space of possible ciphertexts \mathcal{C} : observing a ciphertext $c = \mathcal{E}(m)$ can be thought of as observing an element $c \in \mathcal{C}$ sampled according to Ψ . There is no “model uncertainty” here: as the encryption algorithm \mathcal{E} is public knowledge, all observers know that the “real” distribution Ψ belongs to the parametrized family of probability distributions

$\{\mathcal{E}(m) : m \in \mathcal{M}\}$. Thus, we set the model likelihood function $P(c; m) = \text{Prob}(\mathcal{E}(m) = c)$. In words: given a hypothesized message m and an intercepted ciphertext c , we can compute the likelihood $P(c; m)$ as the probability that the randomized algorithm $\mathcal{E}(m)$ produces c .

The choice of prior distribution $P_{\text{prior}}(m)$ is interesting: a simple choice, reflecting no knowledge at all about what messages are possible, is the uniform distribution on \mathcal{M} . However, this is unrealistic: in a practical context, there is lots of information possible about the range of possible messages. For example, in a modern computing context, the message might be known to be a valid IP packet or a properly-formatted email. Or the message might be known to contain human-generated valid English text. Thus, a savvy eavesdropper could make an informed choice of prior: maybe $P(m)$ is taken as the probability assigned by a highly sophisticated large language model such as GPT-3. In the historical context during which Shannon, Turing, *et al.* developed the foundational concepts of information theory, computing, and cryptography, this sort of statistical knowledge formed the basis of the cryptographic attacks that eventually broke the German World War II Enigma cipher.

From the choice of prior $P_{\text{prior}}(m)$, together with knowledge of the encryption algorithm $\mathcal{E}(m)$, it is then possible to use the Monte Carlo method to estimate the marginal probability $P(c) = \mathbf{E}[P(c; m) | m \sim P_{\text{prior}}(m)]$ by repeated sampling. In simple cases, including our IBE system, an exact answer can be found. For example, if $P_{\text{prior}}(m)$ is uniform on the message space, then $P(c)$ is uniform on the ciphertext space.

Given a sophisticated prior $P_{\text{prior}}(m)$, the posterior probability $P_{\text{posterior}}(m; c)$ may contain much more information about m (in statistical language, this is phrased by stating the distribution $P_{\text{posterior}}(*; c)$ has low entropy) than expected. However, such an attack is *only possible when there is a correlation between the ciphertext and message*: if $P(c; m)$ does not depend on m , then

$$P(c) = \sum_{m \in \mathcal{M}} P(c; m) P_{\text{prior}}(m) = P(c; m_0) \sum_{m \in \mathcal{M}} P_{\text{prior}}(m) = P(c; m_0)$$

for any m_0 . Thus, $P_{\text{posterior}}(m; c) = P_{\text{prior}}(m)$: given c , we learn nothing new about m .

This leads to Shannon's definition of *perfect security*: an encryption algorithm $\mathcal{E}(m)$ is perfectly secure if the distribution of the random variable $\mathcal{E}(m)$ over the random choices made in the course of computing the algorithm does not depend on m . A consequence of this is that for any *predicate* (function valued in $\{0, 1\}$) ϕ on the ciphertext space and any two messages m_0, m_1 , we have

$$\mathbf{E}(\phi(\mathcal{E}(m_0))) = \mathbf{E}(\phi(\mathcal{E}(m_1))) \quad (2.2.2)$$

Shannon proved that the *one-time pad* is a perfectly secure cipher: taking $\{0, 1\}^L$ for the message space and thinking of it as a vector space over \mathbf{F}_2 , the one-time pad is defined by

$$\mathcal{E}(m) = m \oplus \kappa,$$

with \oplus denoting vector addition and κ a key drawn uniformly at random from $\{0, 1\}^L$.

However, the definition of perfect security is too strict to be practically useful. For example, if we consider our “message” to be an RSA secret key p, q and the “ciphertext” to be the corresponding public key $N = pq$, we have

$$P(c = N; m = (p, q)) = \begin{cases} 1 & N = pq \\ 0 & \text{else} \end{cases}$$

Information-theoretically, the integer N perfectly determines its prime factorization! However, RSA is still considered to be cryptographically secure. The divergence from the information-theoretic notion of security comes with the idea of *efficient computability*: effectively, we define *computational security* by requiring (2.2.2) to hold, but only for pairs m_0, m_1 and predicates ϕ which are *efficiently computable*. As there are no known computational problems such that a solution may be efficiently checked but where finding a solution is provably impossible by an efficient (polynomial-bounded) algorithm (i.e. we do not know whether P is equal to NP), results on computational security are always relative: we show that a system satisfies computational security *assuming some underlying mathematical problem is not efficiently computable*.

To accommodate more complicated cryptographic protocols involving multiple rounds of communication, the possibility of dynamic interaction between an eavesdropper and the encryption algorithm, and the like, we instead model security using the notion of an “attack game”: see Section 2.5 for an example in the context of identity-based encryption.

2.3 Identity-based encryption

We will now formally define identity-based encryption, following [2]:

Definition 2.3.1 (Identity-based encryption). An *identity-based encryption scheme* is specified by four (probabilistic polynomial time) algorithms:

- **SETUP**(1^λ) produces the public parameters of the scheme (**params**) and the *master secret key* (**master_key**). The public parameters include the specification¹ of a *message space* \mathcal{M} , a *key space* \mathcal{K} , a *ciphertext space* \mathcal{C} , and an *identity space* \mathcal{ID} (e.g. something like a collection of email addresses), along with any other information which is to be known to all parties and

¹The formal specification of a “space” should include an encoding of elements of that space as strings of bits, as well as efficient algorithms to recognize whether a given string is an element of that space. This is not meant as a constraint on the distribution of elements of the space. For example, we might take the ciphertext space \mathcal{C} to consist of all bit-strings of a given length, perhaps with the first few bits required to be a label, such as an encoding of the string “ciphertext”. It might be the case that not every element of this space is the encryption of a valid message, but we do *not* want an efficient algorithm to be able to distinguish these elements! On the other hand, if we take our ciphertext space to consist of elements of $(\mathbf{Z}/N\mathbf{Z})^\times$, the formal specification should include the provision of an efficient algorithm to determine whether an integer modulo N is coprime to N . We should not work with something like the set of all satisfiable boolean formulas!

needed to perform the other algorithms. For example, these parameters might include a choice of hash function (as in Definition 2.5.2), a choice of elliptic curve, any “public key” associated with `master_key`, etc.

- $\text{EXTRACT}(\text{params}, \text{master_key}, \text{id})$, for $\text{id} \in \mathcal{ID}$, produces the *secret key* $\text{sk}_{\text{id}} \in \mathcal{K}$ associated with identity id .
- $\text{ENCRYPT}(\text{params}, \text{id}, M)$ for $\text{id} \in \mathcal{ID}$ and $M \in \mathcal{M}$ produces the encryption of M to identity id , an element $C \in \mathcal{C}$ called the *ciphertext*.
- $\text{DECRYPT}(\text{params}, \text{id}, s, C)$ for $\text{id} \in \mathcal{ID}, s \in \mathcal{K}, C \in \mathcal{C}$ produces an element of \mathcal{M} . These algorithms are required to be related by the following “soundness” or “consistency” property: for any $M \in \mathcal{M}$ and any $\text{id} \in \mathcal{ID}$, we have

$$\text{DECRYPT}(\text{params}, \text{id}, \text{sk}_{\text{id}}, \text{ENCRYPT}(\text{params}, \text{id}, M)) = M \quad (2.3.1)$$

Here, $(\text{params}, \text{master_key}) = \text{SETUP}(1^\lambda)$ and $\text{sk}_{\text{id}} = \text{EXTRACT}(\text{params}, \text{master_key}, \text{id})$

2.4 RSA Keys

The *RSA* cryptosystem is among the more widely-used cryptographic tools in the world. It is based on the difficulty of factoring integers. To be precise, we have:

Definition 2.4.1. The *RSA master key setup* algorithm $\text{RSA_KEY_GEN}(1^\lambda)$ chooses at random prime numbers p, q , each of bit-length approximately $\lambda/2$, and yields the *master RSA public key* $N = pq$ and *master RSA secret key* (p, q) . Note that N has bit-length approximately equal to λ .

By the Prime Number Theorem, to find a prime p with $\log(p)$ approximately equal to $\lambda/2$, we can test random integers with $\lambda/2$ bits and after approximately $\lambda/2$ tries, we will find a prime. Testing whether a number is a prime with the Miller-Rabin test has expected runtime $O(\lambda^2 \log \log \lambda)$; so overall, the expected runtime of RSA_KEY_GEN is cubic in λ .

We note that here, λ is *not* the actual security level: an RSA modulus of bit-length λ has security level proportional to $\approx 2 \cdot \lambda^{1/3} * (\log \lambda)^{2/3}$ using the general number field sieve algorithm (see [12, Chapter 10]). The NIST recommendation for modern commercial applications is to take $\lambda \geq 2048$, corresponding to a security level of approximately 112 bits. A modern commodity laptop can find a suitable key of this length in around 10 seconds.

2.5 Definitions of security notions

Definition 2.5.1. An identity-based encryption scheme is *adaptive chosen plaintext* - or IND-ID-CPA - *secure* if for any efficient adversary \mathcal{A} , the advantage of \mathcal{A} in the following attack game is

negligible:

- The challenger \mathcal{C} takes the security parameter λ and runs $\text{SETUP}(1^\lambda)$, generating public parameters params and a master secret key master_key . It gives params to the adversary.
- The adversary issues a series of *identity queries*, in which it sends the challenger some $\text{id}_i \in \mathcal{ID}$, and the challenger responds with $\text{EXTRACT}(\text{id}_i)$, the secret key corresponding to id_i .
- The adversary then produces a pair m_0, m_1 of equal-length messages as well as a “challenge identity” id , with the constraint that id is not equal to any of the id_i queried in the previous phase.
- The adversary may make additional identity queries on identities (id_j) so long as $\text{id}_j \neq \text{id}_i$.
- The challenger samples a secret bit $b \in \{0, 1\}$ and sends the adversary the ciphertext $\text{ENCRYPT}(m_b, \text{id})$.
- The adversary responds with a bit \hat{b} , winning if $\hat{b} = b$.

In brief, an IBE scheme is IND-ID-CPA secure if for all probabilistic polynomial time adversary algorithms \mathcal{A} , the probability that $\hat{b} = b$ after running the above game is $1/2 + O(1/\lambda^k)$ for all k .

A ubiquitous ingredient in cryptographic protocols is that of a *hash function*:

Definition 2.5.2. A *hash function* is an efficiently computable function that takes an arbitrary-length piece $x \in \{0, 1\}^*$ of data as input, and outputs a “sufficiently random” element $\mathcal{H}(x) \in \mathcal{D}$ for some fixed finite set \mathcal{D} (such as $\{0, 1\}^{256}$).

There are various ways of attempting to make this definition more precise, corresponding to specific desirable “randomness” properties. However, there are no functions which are known to have these properties, even conditionally on the hardness of standard cryptographic problems such as integer factorization. On the other hand, there are many functions (one very widely used example is SHA256) which empirically appear to satisfy this definition. (If someone were able to break the randomness properties of SHA256, they would be able to steal all the world’s bitcoin; as this has not happened, we can conclude that SHA256 is “sufficiently random” for practical purposes.

For example, a rather strong way to express the idea of a “sufficiently random hash function” is to require that it should be difficult for a computationally bounded algorithm to distinguish between the output of \mathcal{H} on any chosen sequence of pairwise distinct inputs and the result of randomly sampling elements of \mathcal{D} (independently and from the uniform distribution).

A weaker randomness requirement is that it should be computationally infeasible to find *hash collisions*, i.e. pairs of inputs m_0, m_1 with $\mathcal{H}(m_0) = \mathcal{H}(m_1)$. Hash functions satisfying this latter property are called *collision-resistant*. This allows $\mathcal{H}(m)$ to serve as a brief “digest” of an arbitrarily long input m : to verify the claim that some m' is equal to m , one can instead verify that their hashes are equal.

While there are many hash functions for which no known attacks exist which can find some “non-random” property of the output, formally proving that the output of a hash function is indistinguishable from random noise (including various choices of how to make this notion precise) remains a difficult open problem.

Even working purely theoretically, it can be difficult to determine exactly which randomness properties of a hash function are needed to make a security proof work. Therefore, it is a common technique in the cryptography literature to replace hash functions with an idealized model: *random oracles*. This works as follows:

Definition 2.5.3. Given a cryptographic protocol \mathcal{P} which incorporates a hash function $\mathcal{H}: \{0, 1\}^* \rightarrow \mathcal{D}$, we say that \mathcal{P} is secure with respect to some attack game \mathcal{G} (e.g. the IND-ID-CPA game described above) in the *random oracle model* if it is secure for \mathcal{G} when \mathcal{H} is modeled as a *random oracle*:

- Over the course of \mathcal{G} , whenever any party runs an algorithm of \mathcal{P} , any evaluation $\mathcal{H}(q)$ of the hash function with input q is replaced by a *query* to the random oracle \mathcal{O} on the value q .
- The random oracle \mathcal{O} maintains a list of query-response pairs $\mathcal{Q} = \{(q_i, r_i)\}$. Whenever a query q is made to \mathcal{O} , \mathcal{O} responds by sending r_i if $q = q_i$ for some $(q_i, r_i) \in \mathcal{Q}$. Otherwise, it samples $r \xleftarrow{R} \mathcal{D}$ uniformly at random, adds (q, r) to \mathcal{Q} , and responds with r .
- The adversary is allowed to make arbitrary queries to \mathcal{O} (since the adversary must run in polynomial time with respect to λ , the total number of such queries $O(\lambda^k)$ for some k).

Intuitively, we can think of the random oracle \mathcal{O} as a “uniformly random function” from $\{0, 1\}^*$ to \mathcal{D} . The only problem with using this as a definition is that the input space $\{0, 1\}^*$ is infinite, so there is no uniform probability measure on the set of such functions.

Chapter 3

The abstract IBE construction

In this chapter, we discuss our meta-construction of identity-based encryption. As with Cocks' scheme, the master public/secret key pair will be an instance of the RSA factorization problem as in Definition 2.4.1.

We will abstract the algebra needed to construct our generalized Cocks IBE scheme and prove a security reduction in this generality. We defer further discussions of efficient computability of the various ingredients to the scheme until Chapter 4, where we will discuss how to concretely realize this scheme.

3.1 Algebraic setup

First, we introduce a general algebraic setup. The construction of the encryption scheme and proof of its security works in this generality. However, we will see in Chapter 4 that finding *efficient* encryption and decryption algorithms will impose subtle arithmetic constraints.

In addition to providing the foundation for potential future work that goes beyond the scope of this thesis, one reason to use this general language is to understand how far we can stretch Cocks' construction, isolating the key arithmetic features of quadratic residues which make the construction work.

Let A be a finite étale algebra over $\mathbf{Z}/N\mathbf{Z}$ such that both fibers $A_p := A/pA$ and $A_q := A/qA$ are nonzero and of the same rank r over $\mathbf{F}_p, \mathbf{F}_q$ respectively. The *master secret key* consists of the decomposition of A as a product of fields:

$$A = A_p \times A_q = \prod_{\mathfrak{p} | pA} A/\mathfrak{p} \times \prod_{\mathfrak{q} | qA} A/\mathfrak{q} \quad (3.1.1)$$

Here, the notation $\mathfrak{p} | pA$ (resp. $\mathfrak{q} | qA$) means that $\mathfrak{p}, \mathfrak{q}$ are prime ideals of A such that $\mathfrak{p} \cap \mathbf{Z}/N\mathbf{Z} = (p)$ (resp. $\mathfrak{q} \cap \mathbf{Z}/N\mathbf{Z} = (q)$). The fields A/\mathfrak{p} (resp. A/\mathfrak{q}) are finite field extensions of \mathbf{F}_p (resp. \mathbf{F}_q). We

write $\mathbf{F}_p := A/\mathfrak{p}$, $\mathbf{F}_q := A/\mathfrak{q}$ for these fields.

Let G, H be smooth, fiberwise connected, commutative affine group schemes over $\mathrm{Spec}(A)$, and let $\varphi: G \rightarrow H$ be an isogeny (i.e. a finite surjective homomorphism) with *constant* kernel \underline{C} , where C is a finite cyclic group of *prime* order ℓ . We require ℓ to be coprime to N . In a diagram:

$$0 \rightarrow \underline{C} \rightarrow G \xrightarrow{\varphi} H \rightarrow 0 \quad (3.1.2)$$

Note that a fixed choice of isomorphism $\underline{C} \xrightarrow{\sim} \ker(\varphi)$ of A -groups is part of the data included in the algebraic setup.

The significance of our running assumption that the order ℓ of C is prime lies in the following fact:

Proposition 3.1.1. *Let C be a cyclic group of prime order, and let P be a \underline{C} -torsor over a field \mathbf{F} . Then exactly one of the following two possibilities occurs:*

1. *The torsor P is split completely, i.e. there is a C -equivariant \mathbf{F} -isomorphism $\underline{C} \xrightarrow{\sim} P$. In particular, as a scheme over \mathbf{F} , P is isomorphic to a disjoint union of ℓ copies of $\mathrm{Spec}(\mathbf{F})$, one for each point $p \in P(\mathbf{F})$.*
2. *The torsor P is connected, with P isomorphic over \mathbf{F} to $\mathrm{Spec}(\mathbf{F}')$ for a field extension \mathbf{F}'/\mathbf{F} . This field extension is Galois with Galois group isomorphic to C , with the C -action on P corresponding to the Galois action on \mathbf{F}' .*

Proof. Indeed, a \underline{C} -torsor over a field \mathbf{F} is classified by an element of $H^1(\mathbf{F}, \underline{C}) = \mathrm{Hom}(\Gamma_{\mathbf{F}}, C)$, with $\Gamma_{\mathbf{F}}$ the absolute Galois group of \mathbf{F} . Since ℓ is prime, the cocycle $\sigma: \Gamma_{\mathbf{F}} \rightarrow C$ corresponding to the torsor P is either trivial or surjective. The case that it is trivial corresponds to Case 1, and the case that it is surjective corresponds to Case 2. \square

We will use this fact in an essential way in our proof of our main result, Theorem 3.8.2, where we reduce the security of our IBE scheme to the assumed difficulty of the problem of determining whether elements of $H(A)$ are in the image of φ on $G(A)$. (See Section 3.4 for a precise statement of the problem). Another way of thinking about the relevance of the assumption that ℓ is prime is that this assumption assures any non-trivial element of C is a generator. Thus, we can avoid the need to consider intermediate cases in our arguments.

Remark 3.1.2. We expect that one could extend our results without terrible difficulty to the case of general ℓ , using the case of prime ℓ as a “base case” and working inductively: with our general abstract-algebraic formulation, “functorial” properties of our IBE construction should be transparent. Then, as any isogeny with constant kernel can be written as a composition of isogenies of prime order, properties established for prime ℓ should readily extend to the case of arbitrary ℓ .

For example, if one wished to build an IBE scheme based on the problem of detecting 4-th powers in $(\mathbf{Z}/N\mathbf{Z})^\times$, one could break this problem into two steps: distinguishing 4-th powers among squares,

and then distinguishing squares among general elements. Then one could apply our results for squares twice to demonstrate the security of such a system. We note that the proof of Theorem 3.8.2 is “elementary” in the sense that it only relies on general properties of algebraic groups, finite fields, etc.; the more subtle arithmetic input only appears in Chapter 4.

However, we do not attempt to carry out such an argument here.

3.2 Defining the symbol

Now, we can take étale cohomology (which is really just Galois cohomology over the factor fields of A , as we will discuss shortly) to get the following exact sequence:

$$0 \rightarrow \bigoplus_{\mathfrak{r} \in |\mathrm{Spec}(A)|} C \rightarrow G(A) \rightarrow H(A) \xrightarrow{\delta} H_{\mathrm{\acute{e}t}}^1(\mathrm{Spec}(A), C) \rightarrow 0. \quad (3.2.1)$$

Here, we use the notation $|\mathrm{Spec}(A)|$ to refer to the *set* of prime ideals of A . Note that since G is smooth and fiberwise connected, Lang’s theorem tells us that $H_{\mathrm{\acute{e}t}}^1(\mathrm{Spec}(A), G) = 0$, so the connecting map δ in (3.2.1) is indeed surjective.

Moreover, the master secret key - the factorization (3.1.1) - can equivalently be thought of as knowledge of an isomorphism

$$\mathrm{Spec}(A) \xrightarrow{\sim} \mathrm{Spec}(A_p) \sqcup \mathrm{Spec}(A_q) \xrightarrow{\sim} \bigsqcup_{\mathfrak{p} | pA} \mathrm{Spec}(\mathbf{F}_{\mathfrak{p}}) \sqcup \bigsqcup_{\mathfrak{q} | qA} \mathrm{Spec}(\mathbf{F}_{\mathfrak{q}}) \xrightarrow{\sim} \bigsqcup_{\mathfrak{r} \in |\mathrm{Spec}(A)|} \mathrm{Spec}(\mathbf{F}_{\mathfrak{r}}) \quad (3.2.2)$$

The exact sequence (3.2.1) is a direct sum of analogous sequences with A replaced by each factor field $\mathbf{F}_{\mathfrak{r}}$ of A , via (3.2.2). For a prime ideal \mathfrak{r} of A , we have an exact sequence:

$$0 \rightarrow C \rightarrow G(\mathbf{F}_{\mathfrak{r}}) \rightarrow H(\mathbf{F}_{\mathfrak{r}}) \rightarrow H^1(\mathbf{F}_{\mathfrak{r}}, C) \rightarrow 0 \quad (3.2.3)$$

This H^1 can be computed as Galois cohomology:

$$H^1(\mathbf{F}_{\mathfrak{r}}, C) \xrightarrow{\sim} \mathrm{Hom}(\Gamma_{\mathfrak{r}}, C) \xrightarrow[F]{\sim} C \quad (3.2.4)$$

Here, $\Gamma_{\mathfrak{r}}$ denotes the absolute Galois group of the finite field $\mathbf{F}_{\mathfrak{r}}$. This Galois group is *canonically* isomorphic to $\widehat{\mathbf{Z}}$, with generator the (arithmetic) Frobenius automorphism $\mathrm{Frob}_{\mathfrak{r}}: x \mapsto x^{|\mathfrak{r}|}$, where $|\mathfrak{r}|$ refers to the cardinality of $\mathbf{F}_{\mathfrak{r}}$. Thus, the map labelled F in (3.2.4) is canonical, consisting of evaluation at the Frobenius automorphism.

Combining this isomorphism with (3.2.3), we obtain an isomorphism

$$\mathrm{symb}_{\mathfrak{r}}: H(\mathbf{F}_{\mathfrak{r}})/\varphi(G(\mathbf{F}_{\mathfrak{r}})) \xrightarrow{\sim} C$$

Let's spell out how to compute $\text{symb}_{\mathfrak{r}}(h)$ for a given $h \in H(\mathbf{F}_{\mathfrak{r}})$ by unraveling the connecting map δ in (3.2.3). Over the field $\mathbf{F}_{\mathfrak{r}}$, the terms of (3.1.2) can be replaced with modules over the absolute Galois group $\Gamma_{\mathfrak{r}}$ of $\mathbf{F}_{\mathfrak{r}}$:

$$0 \rightarrow C \rightarrow G(\overline{\mathbf{F}}_{\mathfrak{r}}) \rightarrow H(\overline{\mathbf{F}}_{\mathfrak{r}}) \rightarrow 0$$

Here, $\overline{\mathbf{F}}_{\mathfrak{r}} = \cup_{n=1}^{\infty} \mathbf{F}_{\mathfrak{r}^n}$ is the algebraic (or equivalently, separable) closure of $\mathbf{F}_{\mathfrak{r}}$. The Galois group $\Gamma_{\mathfrak{r}}$ acts on $G(\overline{\mathbf{F}}_{\mathfrak{r}})$ and $H(\overline{\mathbf{F}}_{\mathfrak{r}})$ via functoriality from its action on $\overline{\mathbf{F}}_{\mathfrak{r}}$. More explicitly, if we have a closed immersion of G (resp. H) into affine space $\mathbf{A}_{\mathbf{F}_{\mathfrak{r}}}^n$, the Galois action is the natural action on coordinates, with $\sigma \in \Gamma_{\mathfrak{r}}$ sending (x_1, \dots, x_n) to $(\sigma(x_1), \dots, \sigma(x_n))$.

Now, by the assumption that $\varphi: G \rightarrow H$ is an isogeny, so in particular surjective on geometric points, given $h \in H(\mathbf{F}_{\mathfrak{r}})$, we can find some finite extension $\mathbf{F}_{\mathfrak{r}^k}$ of $\mathbf{F}_{\mathfrak{r}}$ and $\tilde{h} \in G(\mathbf{F}_{\mathfrak{r}^k})$ such that $\varphi(\tilde{h}) = h$. By the definition of the connecting map in Galois cohomology, the map from $H(\mathbf{F}_{\mathfrak{r}})$ to $H^1(\mathbf{F}_{\mathfrak{r}}, C)$ in (3.2.3) takes h to the 1-cocycle on $\Gamma_{\mathfrak{r}}$ which maps $\sigma \in \Gamma_{\mathfrak{r}}$ to $\sigma(\tilde{h}) * \tilde{h}^{-1} \in G(\mathbf{F}_{\mathfrak{r}^k})$. As the isogeny φ is defined over $\mathbf{F}_{\mathfrak{r}}$, the Galois action commutes with φ ; in addition, as φ is a group homomorphism, it intertwines the multiplication and inversion operations on G with those on H . Therefore, we have

$$\varphi(\sigma(\tilde{h}) * \tilde{h}^{-1}) = \sigma(\varphi(\tilde{h})) *_H \varphi(\tilde{h})^{-1} = \sigma(h) *_H h^{-1} = h * h^{-1} = 1$$

Here, we have $\sigma(h) = h$, as $h \in H(\mathbf{F}_{\mathfrak{r}})$. Thus, we see that $\sigma(\tilde{h}) * \tilde{h}^{-1}$ is actually an element of $(\ker \varphi)(\mathbf{F}_{\mathfrak{r}^k}) = C$. Finally, we obtain the value of the symbol by evaluating this cocycle at the Frobenius automorphism. Therefore, we have:

$$\text{symb}_{\mathfrak{r}}(h) = \text{Frob}_{\mathfrak{r}}(\tilde{h}) *_G \tilde{h}^{-1} \quad (3.2.5)$$

Example 3.2.1. Suppose that $A = \mathbf{Z}/N\mathbf{Z}$ with $G = H = \mathbf{G}_m$, and $\varphi: \mathbf{G}_m \rightarrow \mathbf{G}_m$ is the squaring map. Then $H(\overline{\mathbf{F}}_{\mathfrak{r}}) = G(\overline{\mathbf{F}}_{\mathfrak{r}}) = \overline{\mathbf{F}}_{\mathfrak{r}}^{\times}$. We may identify \mathbf{G}_m with the hyperbola $xy = 1$ inside of affine space $\mathbf{A}_{\mathbf{F}_{\mathfrak{r}}}^2$ by the map $x \mapsto (x, x^{-1})$. Then, we see that the action of $\Gamma_{\mathfrak{r}}$ on $\overline{\mathbf{F}}_{\mathfrak{r}}^{\times}$ is just the usual Galois action. Therefore, $\text{Frob}_{\mathfrak{r}}(\tilde{h}) = \tilde{h}^{\mathfrak{r}}$. Moreover, $\tilde{h} \in \overline{\mathbf{F}}_{\mathfrak{r}}^{\times}$ is defined to be an element such that $\tilde{h}^2 = \varphi(\tilde{h}) = h$, so it is a square root of h . As N is odd, we see that $\mathfrak{r} - 1$ is divisible by 2. Thus, we may compute:

$$\text{symb}_{\mathfrak{r}}(h) = \tilde{h}^{\mathfrak{r}-1} = \left(\tilde{h}^2\right)^{\frac{\mathfrak{r}-1}{2}} = h^{\frac{\mathfrak{r}-1}{2}} = \left(\frac{h}{\mathfrak{r}}\right)$$

Thus, we see that the symbol in this case is the Legendre symbol.

Returning to $\text{Spec}(A)$, we can combine the local symbols to define:

Definition 3.2.2. Given a short exact sequence $0 \rightarrow \underline{C} \rightarrow G \xrightarrow{\varphi} H \rightarrow 0$ as in (3.1.2), we define the *symbol*

$$\text{symb}_{\varphi}: H(A) \rightarrow C$$

as

$$\text{symb}_\varphi(h) = \prod_{\mathfrak{r} \in |\text{Spec}(A)|} \text{symb}_{\mathfrak{r}}(h_{\mathfrak{r}}) \quad (3.2.6)$$

where $h = (h_{\mathfrak{r}})_{\mathfrak{r} \in |\text{Spec}(A)|} \in H(A) \simeq \prod_{\mathfrak{r} \in |\text{Spec}(A)|} H(\mathbf{F}_{\mathfrak{r}})$ is the decomposition of h according to (3.1.1).

Example 3.2.3. Returning to Example 3.2.1, with $H = G = \mathbf{G}_m$, $A = \mathbf{Z}/N\mathbf{Z}$, and φ the squaring map, we see that $|\text{Spec}(A)| = \{p, q\}$, and

$$\text{symb}_\varphi(h) = \left(\frac{h}{p}\right) * \left(\frac{h}{q}\right) = \left(\frac{h}{N}\right)$$

Thus, symb_φ in (3.2.6) generalizes the Jacobi symbol. We follow this analogy closely, and our IBE scheme uses symb_φ precisely how the Cocks scheme uses the Jacobi symbol.

3.3 The master key

Now, we have set up enough of the general context that we can describe the first of the four algorithms comprising our IBE scheme, collecting the information needed to be formally specified in order to implement our scheme and which will be needed to be public knowledge in order to encrypt or decrypt messages. Primarily, this consists of efficiently computable representations of the algebraic objects we have discussed so far. There are, however, two additional pieces of data that must be revealed as public knowledge in order for encryption and decryption to be possible:

First, we emphasize that the definition of the symbol symb_φ in Section 3.2 is *not* intrinsic to the isogeny φ between algebraic groups over A : of crucial importance to our definition was the isomorphism (of A -group schemes) between \underline{C} and the kernel of φ . Let c_0 be a generator of the cyclic group C . Then, this isomorphism $\underline{C} \xrightarrow{\sim} (\ker \varphi)(A)$ amounts to the choice of a global section

$$\zeta = (\zeta_{\mathfrak{r}})_{\mathfrak{r}} \in (\ker \varphi)(A) \simeq \prod_{\mathfrak{r} \in |\text{Spec}(A)|} (\ker \varphi)(\mathbf{F}_{\mathfrak{r}}),$$

where ζ is the image under this isomorphism of the constant function $|\text{Spec}(A)| \rightarrow C$ with value c_0 . In terms of this ζ , the isomorphism $\underline{C} \xrightarrow{\sim} (\ker \varphi)(A)$ takes a function $|\text{Spec}(A)| \rightarrow C$, $\mathfrak{r} \mapsto (c_0)^{c_{\mathfrak{r}}}$, with $c_{\mathfrak{r}} \in \mathbf{Z}/\ell\mathbf{Z}$, to the section $(\zeta_{\mathfrak{r}}^{c_{\mathfrak{r}}})_{\mathfrak{r} \in |\text{Spec}(A)|}$. Note that replacing our choice of c_0 with some other generator $c_1 = c_0^a \in C$, $a \in (\mathbf{Z}/\ell\mathbf{Z})^\times$, results in replacing ζ by ζ^a : here, a is a *constant* value. The significance of this choice is that it is what allows us to consistently multiply the local symbols together: *a priori*, $\text{symb}_{\mathfrak{r}}$ is only valued in $(\ker \varphi)(\mathbf{F}_{\mathfrak{r}})$ for varying \mathfrak{r} .

The second subtlety comes from the need to be able to use an element in $\varphi(G(A))$ in the encryption process, *without* knowing the factorization of N . Per our upcoming computational hardness assumption (Problem 3.4.1) it is infeasible to distinguish between elements of $\varphi(G(A))$ and those in

$\ker(\text{symb}_\varphi)$. Thus, we must use the brute-force approach of trying every possibility.

Note that (3.2.1) and (3.2.4) imply that $H(A)/\varphi(G(A)) \simeq \bigoplus_{\mathfrak{r} \in |\text{Spec}(A)|} C$. Restricting to the kernel of symb_φ corresponds to restricting to the set of elements such that the product of their components is 1. Thus, we need $\ell \cdot (n_A - 1)$ coset representatives. In particular, it will be important for the encryption algorithm that given an element $h \in H(A)$, there is exactly one α_i with $\alpha_i h \in \varphi(G(A))$.

Similarly to the quadratic non-residue $u \in (\mathbf{Z}/N\mathbf{Z})^\times$ with $(\frac{u}{N}) = 1$ from Definition 1.3.2, the public parameters for our IBE scheme must include a complete set of coset representatives $\{\alpha_i\}_{i=1}^{\ell(n_A-1)}$ for $\ker(\text{symb}_\varphi)$ modulo $\varphi(G(A))$. Here n_A is the number of prime ideals of A .

Definition 3.3.1 (The **SETUP** algorithm). The *public parameters* **params** of our IBE scheme consist of the following information:

- A prime number ℓ .
- An RSA composite number N whose binary representation has length λ .
- A presentation for a finite étale $\mathbf{Z}/N\mathbf{Z}$ -algebra A with constant rank r , i.e. an isomorphism

$$(\mathbf{Z}/N\mathbf{Z})[x_1, \dots, x_m]/(f_1, \dots, f_k) \xrightarrow{\sim} A.$$

- Presentations for algebraic groups G, H over $\text{Spec}(A)$, i.e. closed immersions into affine spaces over $\text{Spec}(A)$ together with explicit equations for their image as well as explicit polynomial formulas for the multiplication and inversion maps in these coordinates.
- A presentation for an isogeny $\varphi: G \rightarrow H$ in terms of the chosen coordinates on G and H .
- An isomorphism from \underline{C} , with C a cyclic group of order ℓ , to the kernel of φ : if we identify C with $\mathbf{Z}/\ell\mathbf{Z}$ by choosing a generator $c_0 \in C$, such an isomorphism is uniquely specified by the choice of some $\zeta \in (\ker \varphi)(A)$ of fiberwise order ℓ , the image of the constant section with value c_0 .
- $\ell(n_A - 1)$ elements $\{\alpha_i\}_{i=1}^{\ell(n_A-1)}$ of $\ker(\text{symb}_\varphi) \subseteq H(A)$ with $\alpha_1 = 1$ such that $\{\alpha_i\}$ forms a set of coset representatives of $\ker(\text{symb}_\varphi)$ modulo the image $\varphi(G(A))$: n_A is the number of prime ideals of A .

The *master secret key* **master_key** consists of the explicit decomposition of A into a product of fields as in (3.1.1).

The **SETUP** algorithm, on input 1^λ , runs **RSA_KEY_GEN**(1^λ) to choose an RSA key $N = pq$ of key-length λ . The α_i are then sampled at random: knowing the factorization of N , we may compute the factorization of A into prime ideals \mathfrak{r} and then compute $\text{symb}_\mathfrak{r}$ for each $\mathfrak{r} \in \text{Spec}(A)$, so by repeatedly sampling elements of $H(A)$ we can find all of the α_i 's.

A few remarks about computability are in order. The above definition is general, and formally, we allow the **SETUP** algorithm to be an arbitrary probabilistic polynomial time algorithm which can produce parameters meeting the above requirements. For example, in principle, the parameters such as G, H, φ , and A could be selected dynamically as a function of N . However, in our incarnation of this general construction in Chapter 4, and any version of this construction we can imagine, the algebraic objects in **params** come from reducing a “global” version of the setup modulo N . In particular, we will have an exact sequence

$$0 \rightarrow \underline{C} \rightarrow G_0 \xrightarrow{\varphi_0} H_0 \rightarrow 0$$

with G_0, H_0 algebraic groups and φ_0 an isogeny, defined over some finite étale algebra A_0 of rank r over $\mathbf{Z}[1/\Delta]$ for some Δ which is of constant size with respect to λ . Moreover, we will have explicit coordinates for G_0, H_0, A_0 , and φ_0 as required in Definition 4.1.2. Then we obtain the **params** by reducing everything modulo N . In such an arrangement, the **SETUP** algorithm does not need to be able to efficiently compute these parametrizations: the only item that depends on N intrinsically is the set of coset representatives $\{\alpha_i\}$.

However A is chosen, it is crucial that the **SETUP** algorithm can efficiently compute the master secret key using the factorization of N . This is always feasible: given the factorization $N = pq$, one can factor A as $A = A_p \times A_q$, with $A_p = A/pA$, $A_q = A/qA$. Then A_p (resp. A_q) is a finite étale algebra of rank r over \mathbf{F}_p (resp. \mathbf{F}_q). Working over a finite *field*, there are efficient algorithms to factor étale algebras, and in particular to factor polynomials: see [12, §3.4, §6.2.4]).¹ Note that we are considering ℓ and r as small, fixed constants with respect to λ : in particular, we will have $\ell, r \ll \lambda$. Note that in particular, this comment implies that knowledge of the factorization of N is sufficient to compute **master_key**: therefore, we will sometimes abuse notation and refer to this factorization as the master secret key.

3.4 The isogeny image assumption

Next, we discuss the computational problem which will form the basis for the security of our scheme. This is the *isogeny image assumption*: without knowing the factorization of N , it should be impossible to efficiently determine whether an element in $\ker(\text{symb}_\varphi)$ is in the image $\varphi(G(A))$ or

¹However, beware that these algorithms do not strictly satisfy the definition of “efficient algorithm” we gave in Section 2.1: our definition of “probabilistic polynomial time” requires the algorithm to definitely terminate in polynomial time, whereas the algorithms to factor polynomials over a finite field involve random sampling and could fail to terminate. One can choose to either have a hard cap on the runtime with some probability of failing to find a factor, or allow the runtime to be polynomial only in *expectation*. Another option is to assume some version of the Riemann Hypothesis to derive deterministic bounds. The situation is similar for the Miller-Rabin primality test (in that case, the AKS algorithm is unconditionally known to produce the correct result in polynomial time, but is not used in practice, being much slower in expectation than the Miller-Rabin test). Of course, for anything in cryptography to be meaningful, one must assume that P is not equal to NP, so assuming the Riemann Hypothesis is not exactly outlandish. See [16] for a survey of related problems and results.

not.

However, note the key property of this hard problem which makes it suitable for constructing an IBE scheme: *given the master secret key*

$$A \xrightarrow{\sim} \prod_{\mathfrak{r}} \mathbf{F}_{\mathfrak{r}}, \quad (3.4.1)$$

the isogeny image problem can easily be solved. The isomorphism (3.4.1) lets us write $h \in H(A)$ as $(h_{\mathfrak{r}})_{\mathfrak{r}} \in \prod_{\mathfrak{r}} H(\mathbf{F}_{\mathfrak{r}})$: now, h is in the image of φ exactly when

$$\text{symb}_{\mathfrak{r}}(h_{\mathfrak{r}}) = 1$$

for all \mathfrak{r} and is in the kernel of symb_{φ} when

$$\prod_{\mathfrak{r}} \text{symb}_{\mathfrak{r}}(h_{\mathfrak{r}}) = 1$$

Formally, we have the following computational hardness assumption, which generalizes the Quadratic Residuosity Problem 1.2.2:

Problem 3.4.1 (Isogeny image problem). Fix an exact sequence $0 \rightarrow \underline{\mathbb{C}} \rightarrow G \xrightarrow{\varphi} H \rightarrow 0$ of smooth fiberwise connected affine groups over a finite $\mathbf{Z}/N\mathbf{Z}$ -algebra A as above. The *isogeny image problem* consists of the challenge of being able to distinguish, for a randomly chosen element h of $\ker(\text{symb}_{\varphi})$, whether h is in the image of φ . More precisely, an *adversary* \mathcal{A} , which is some probabilistic polynomial time algorithm, competes in the following challenge game:

- A *challenger* (algorithm) \mathcal{C} draws a random secret bit $b \in \{0, 1\}$.
- If $b = 1$, then \mathcal{C} samples² a uniformly random element h from $\varphi(G(A)) \subseteq H(A)$, and if $b = 0$, then \mathcal{C} samples a uniformly random element h from $\ker(\text{symb}_{\varphi}) - \varphi(G(A))$.
- The algorithm \mathcal{C} then feeds h into \mathcal{A} along with the public parameters **params**.
- \mathcal{A} must output a bit \widehat{b} .
- We say that \mathcal{A} wins the game if $\widehat{b} = b$.

For such an adversary algorithm \mathcal{A} , we define the *advantage* of \mathcal{A} as

$$\text{adv}(\mathcal{A})(\lambda) = \left| \text{Prob}(\widehat{b} = b) \right| - 1/2$$

²The challenger is allowed access to secret information, so they can factor A into a product of fields and compute the local symbols to be able to determine when an element of $H(A)$ is in $\varphi(G(A))$. So the sampling algorithm could go as follows: if $b = 1$ choose a random element of $G(A)$ and apply φ , and if $b = 0$ choose a random element $h \in H(A)$ and if either $h \in \varphi(G(A))$ or $\text{symb}_{\varphi}(h) \neq 0$, try again.

With the probability taken over the internal randomness of the probabilistic polynomial time algorithm \mathcal{A} as well as the random choice of the bit b in the game. In other words, the advantage of \mathcal{A} is the difference between its probability of success and the probability of success given by random guessing. We say that *the isogeny image assumption* holds if for all such probabilistic polynomial time adversary algorithms \mathcal{A} , $\text{adv}(\mathcal{A})(\lambda)$ is a negligible function of the security parameter λ (i.e. it is $O(1/\lambda^k)$ for all k .)

In general, given two probability distributions $(\mathcal{E}, \mu), (\mathcal{E}', \mu')$ with $\mathcal{E}, \mathcal{E}'$ finite sets, we can consider a game like the above, with \mathcal{C} choosing a random bit b and sampling from (\mathcal{E}, μ) if $b = 0$ and from (\mathcal{E}', μ') if $b = 1$ and the adversary \mathcal{A} attempting to guess b based on the result. We say these two distributions are *computationally indistinguishable* if any such (probabilistic polynomial time) adversary \mathcal{A} has negligible advantage in this game. The intuition here is that this definition should mean that it is practically impossible to distinguish between the two probability distributions.

With this language, the isogeny image assumption says that the two sets $\ker(\text{symb}_\varphi) - \varphi(G(A))$ and $\varphi(G(A))$ (as always, sampled according to the uniform distribution) are computationally indistinguishable without knowing the factorization of A into a product of fields (in particular, this assumption includes the assumption that it is computationally infeasible to factor N).

Once we finish describing our IBE scheme, we will prove a security reduction to this assumption in Theorem 3.8.2.

3.5 Extracting private keys

Now, we will describe the second algorithm comprising an IBE system: the **EXTRACT** algorithm, which uses the master secret `master_key` to generate private keys based on user identities $\text{id} \in \mathcal{ID}$. The space of identities \mathcal{ID} can be arbitrary data: for example, they might consist of usernames, email addresses, etc. What matters is that they have some canonical encoding as strings of bits.

We're also free to use various key management practices here to enhance the real-life security of the scheme: for example, the ID could be the concatenation of a user's email address with today's date. This achieves some forward secrecy: if a secret key is exposed to a malicious party, that party can only decrypt today's messages and not all messages ever sent to the user.

Another very important practicality, on which we will make no further comment, is the problem of authenticating users to the central trusted party: this is analogous to the problem, ubiquitous in public key cryptography, of verifying ownership of a public key. This is typically managed by a certificate authority. For an in-depth discussion of "real-life" identity-based encryption, including discussion of such issues, see [9].

We will need a hash function, known to all parties

$$\mathcal{H}: \mathcal{ID} \rightarrow \ker(\text{symb}_\varphi) \subseteq H(A)$$

For example, such a hash function may be built deterministically by taking \mathcal{H}_0 to be any hash function valued in $H(A)$ and defining $\mathcal{H}(\text{id})$ to be $\mathcal{H}_0(\text{id}||i)$ for the first integer i such that the symbol is 0. Here, “||” means concatenation of bit-strings; e.g. representing i as a 32-bit integer.³

Since \mathcal{H}_0 is a hash function, this process should be computationally indistinguishable from a sequence of random choices of elements of $H(A)$: in particular, each of the ℓ possible values of $\text{symb}_\varphi(h)$ should be equally likely, so this process requires $O(\ell)$ invocations of \mathcal{H}_0 on average. Formally, we model \mathcal{H}_0 as a *random oracle* as in Definition 2.5.3.

We can use \mathcal{H} to define the “public key” associated to an identity id as

$$\text{pk}_{\text{id}} := \mathcal{H}(\text{id}). \quad (3.5.1)$$

Since $\text{pk} \in \ker(\text{symb}_\varphi)$, there is exactly one α_i from the **params** such that $\alpha_i * \text{pk} \in H(A)$. As knowledge of **master_key** is sufficient to solve the isogeny image Problem 3.4.1, the central trusted party can find this α_i . Now, we can define the private key for identity id :

Definition 3.5.1. For identity $\text{id} \in \mathcal{ID}$, the central trusted party determines the secret key for identity id by running

$$\text{EXTRACT}(\text{params}, \text{master_key}, \text{id}) = (\alpha_i, \text{sk}_{\text{id}}) : \text{sk}_{\text{id}} \in \varphi^{-1}(\text{pk}_{\text{id}} * \alpha_i)$$

where sk_{id} is chosen uniformly at random from

$$\varphi^{-1}(\text{pk}_{\text{id}} * \alpha_i) = \{y \in G(A) \mid \varphi(y) = \text{pk}_{\text{id}} * \alpha_i\}$$

We note that given **master_key** as well as the information encoded in **params** (Definition 3.3.1), it is always possible to solve for preimages. Indeed, for $\mathfrak{r} \in \text{Spec}(A)$, the problem of finding a preimage in $G(\mathbf{F}_{\mathfrak{r}})$ for some $h \in H(\mathbf{F}_{\mathfrak{r}})$ is equivalent to splitting the rank- ℓ étale $\mathbf{F}_{\mathfrak{r}}$ -algebra $G \times_{\varphi, H, h} \text{Spec}(\mathbf{F}_{\mathfrak{r}})$. From the information in **params**, we may obtain an explicit encoding of this algebra. There are efficient algorithms that solve this problem in general for étale algebras of fixed rank over a finite field, using linear algebraic methods: see [12, §3.4, §6.2.4].

3.6 Encryption

We are now ready to describe the encryption algorithm: let’s say that Alice wants to send a message m to Bob, whose identifier is id_B . Knowing id_B , she may compute Bob’s “public key”:

$$\text{pk}_B = \mathcal{H}(\text{id}_B)$$

³Such encoding details are not important for our purposes: all that matters is that there is some deterministic, agreed-upon method to generate elements of $\ker(\text{symb}_\varphi)$ in such a way that the output is “pseudo-random”, i.e. there is no useful relationship between the value of id and $\mathcal{H}(\text{id})$.

We consider the message space \mathcal{M} to consist of strings of elements of C : if $m \in \mathcal{M}$, we have $m = c_1 c_2 c_3 \dots c_L$ with $c_i \in C$: thus, we may think of m as an element of $C^{\oplus L}$.

We will describe the encryption algorithm in terms of a randomized algorithm $\mathcal{E}(m, v)$, for $v \in \ker(\text{symb}_\varphi)$, with the property that m may be recovered from $\mathcal{E}(m, v)$ using a preimage $g \in G(A)$ with $\varphi(g) = v$, should one exist. At the same time, recovering any information about m from $\mathcal{E}(m, v)$ without knowing such a preimage should be approximately as hard as solving the isogeny image Problem 3.4.1. (We'll formulate this security notion precisely in Section 3.8.)

Alice knows that pk_B is in $\ker(\text{symb}_\varphi)$, but without the master key, she cannot determine its image in $\ker(\text{symb}_\varphi) / \varphi(G(A))$. In other words, she knows that there is exactly one α_i from **params** such that $\alpha_i * \text{pk}_B$ is in $\varphi(G(A))$, but she cannot determine the i for which this is true. Therefore, she must account for every possibility, preparing $\ell * (n_A - 1)$ distinct ciphertexts, one for each of the α_i from **params**. All but one of these will be useless, containing no information about m in a strong sense, as we will see in Section 3.8: the ciphertext for each α_i is encoded in the hope that $\alpha_i * \text{pk}_B \in \varphi(G(A))$, and the message can only be recovered if this is true. With this notation, the final ciphertext will be

$$(\mathcal{E}(m, \alpha_i * \text{pk}_B))_{i=1}^{\ell * (n_A - 1)} \quad (3.6.1)$$

As with the Cocks IBE scheme from Section 1.3, Alice encrypts her message one character at a time. If we like, we can think of the encryption algorithm as applying a mod- ℓ version of the “one-time pad”: to obtain the ciphertext $\mathcal{E}(m, v)$, Alice generates a “keystream” $\kappa = (k_1, k_2, \dots, k_L) \in C^L$ which she can then component-wise multiply with $m \in C^L$. Along with this product, her ciphertext must include some additional information: a “clue” $d_i \in H(A_{\varphi, v})$, one for each character c_i . Here, $A_{\varphi, v}$ will be a finite extension of A which depends on v . (We'll describe this in more detail presently.)

So we may write:

$$\mathcal{E}(m, v) = (\kappa \cdot m, \delta) = (k_1 * c_1, k_2 * c_2, \dots, k_L * c_L, d_1, d_2, \dots, d_L)$$

If Bob's secret key sk_B is in $\varphi^{-1}(v)$, he needs to be able to compute κ from δ along with sk_B , but it should be computationally infeasible for an eavesdropper Eve to be able to compute κ (with accuracy non-negligibly better than random guessing) from $(\kappa \cdot m, \delta)$ and pk_B .

To choose κ , Alice will sample an auxiliary value $\tau = (t_1, t_2, \dots, t_L) \xleftarrow{R} H(A)^L$, i.e. drawing L independent and uniformly distributed elements from $H(A)$. Then she defines

$$\kappa = \text{symb}_\varphi(\tau) = (\text{symb}_\varphi(t_1), \dots, \text{symb}_\varphi(t_L)) \quad (3.6.2)$$

Note that κ is uniformly distributed in C^L : symb_φ is a *group homomorphism* from $H(A)$ to C , which implies that there are equally many elements of $H(A)$ mapping to each $c \in C$ (as they are cosets of $\ker(\text{symb}_\varphi)$). Thus, if τ is drawn from a uniform distribution on $H(A)^L$, the random variable $\kappa = \text{symb}_\varphi(\tau)$ is uniformly distributed on C^L .

Next, we describe the nature of Alice’s “clue” δ : this will encode $\tau \in H(A)^L$ modulo $\varphi(G(A))^L$, but only for someone who knows a preimage of v .

To do this, we pass to the scheme

$$\text{Spec}(A_{\varphi,v}) := \varphi^{-1}(v) = G \times_{\varphi, H, v} \text{Spec}(A).$$

Here, $A_{\varphi,v} = A[G]/\varphi^{-1}(I_v)$ where $G = \text{Spec}(A[G])$ and I_v is the kernel of the A -algebra map $A[G] \rightarrow A$ determined by v (“evaluation at v ”). This is a \underline{C} -torsor over $\text{Spec}(A)$: in particular, it is an étale A -algebra.

Note that if $v = \varphi(g)$, this torsor is *split* over A : there exists an A -algebra isomorphism

$$A_{\varphi,v} \xrightarrow{\sim} \prod_{j=1}^{\ell} A \quad (3.6.3)$$

with the map given by the product of the A -algebra maps $A[G] \rightarrow A$ associated to the ℓ elements $g \cdot \zeta^j \in G(A)$, each satisfying $\varphi(g \cdot \zeta^j) = v$. In particular, the knowledge of such a section $A[G] \rightarrow A$ is equivalent to the knowledge of a preimage of v .

Now, we may define $\delta = (d_1, \dots, d_L)$: for each i , d_i is an element of $H(A_{\varphi,v})$ obtained by multiplying t_i by a random element of $\varphi(G(A_{\varphi,v}))$: Alice samples some $\beta = (b_1, \dots, b_L) \xleftarrow{R} G(A_{\varphi,v})^L$, and defines

$$\delta := \varphi(\beta) * \tilde{\tau} \in H(A_{\varphi,v})^L. \quad (3.6.4)$$

Here, $\tilde{\tau}$ is the image of τ under the “restriction” map $H(A)^L \rightarrow H(A_{\varphi,v})^L$.

In other words, she is obfuscating the value of $\tau \in H(A)^L$ by first embedding it inside $H(A_{\varphi,v})^L$ and then multiplying it with a random element which is in the image of φ in *this bigger ring*. The purpose of working in this bigger ring is that Bob, armed with the splitting (3.6.3), can recover $\kappa = \text{symb}_{\varphi}(\tau)$, but an eavesdropper cannot.

Thus, in sum, we have:

Definition 3.6.1. To encrypt a message m to the user with identifier id_B , Alice runs:

```

for  $i = 1, \dots, \ell * (n_A - 1)$  do
     $v = \mathcal{H}(\text{id}_B) * \alpha_i$ ;
     $\tau \xleftarrow{R} H(A)^L$ ;
     $\beta \xleftarrow{R} G(A_{\varphi,v})^L$ ;
     $\kappa = \text{symb}_{\varphi}(\tau)$ ;
     $\delta = \varphi(\beta) * \tilde{\tau}$ ;
     $\mathcal{C}_i = \mathcal{E}(m, v) = (\kappa \cdot m, \delta)$ 
end
ENCRYPT $(m, \text{id}_B) = (\mathcal{C}_1, \dots, \mathcal{C}_{\ell(n_A-1)})$ ;

```


Here, the “=” signs are the “declarative =”, meaning that we set the variable on the left of the = equal to the value of the expression on the right. The notation “ $x \xleftarrow{R} \mathcal{X}$ ” refers to drawing a sample x from the uniform distribution on the set \mathcal{X} , as defined in Section 2.1.

Note: If we want to implement this scheme, there’s a glaring obstacle: it’s not at all clear how a user of the IBE system can compute symb_φ without knowing the master secret key! Indeed, this will be our primary design constraint, as well as where class field theory enters the picture: recall that in the Cocks IBE scheme from Section 1.3, the algorithm to compute the Jacobi symbol from Theorem 1.2.1 uses Theorem 1.1.8, the law of quadratic reciprocity, in an essential way. We need to find an appropriate generalization of this remarkable bit of number theory which will apply to more general G, H , and A : fortunately for us, many of the greatest number theorists of the last century spent a great deal of their energy developing exactly such a generalization! We’ll discuss this issue in depth in Chapter 4.

3.7 Decryption

Next, we describe how Bob can decrypt Alice’s message.

Bob receives the ciphertext

$$\text{ENCRYPT}(m, \text{id}_B) = (\mathcal{E}(m, \text{pk}_B * \alpha_1), \mathcal{E}(m, \text{pk}_B * \alpha_2), \dots, \mathcal{E}(m, \text{pk}_B * \alpha_{\ell(n_A-1)}))$$

where $\text{pk}_B = \mathcal{H}(\text{id}_B)$ is Bob’s public key (recall that $\alpha_1 = 1$).

First, Bob discards the irrelevant components: Bob’s secret key $s := \text{sk}_B \in G(A)$ satisfies $\varphi(s) = \text{pk}_B * \alpha_i$ for some i , and Bob knows which one this is. He received his secret key from the **ENCRYPT** algorithm of Section 3.5, whose output includes α_i ; alternatively, he could figure out the α_i himself by computing

$$\alpha_i = \varphi(s) * \mathcal{H}(\text{id}_B)^{-1}$$

Bob then keeps the i -th component of the ciphertext and discards the rest. This component is $\mathcal{E}(m, v)$, where $v := \varphi(\text{sk}_B)$.

The secret key s determines an A -algebra map $\pi_s: A_{\varphi, v} \rightarrow A$ via the projection $A[G] \rightarrow A$ associated to s as in (3.6.3). Indeed, by the fiber product definition of $\text{Spec}(A_{\varphi, v})$, such splittings are equivalent to preimages of v under $\varphi: G(A) \rightarrow H(A)$.

Now, from Definition 3.6.1, we have $\mathcal{E}(m, v) = (\kappa \cdot m, \delta)$, where $\kappa = \text{symb}_\varphi(\tau)$ and $\delta = \varphi(\beta) * \tilde{\tau}$: κ is the “key” and δ is the “clue” from Section 3.6.

To decrypt, Bob applies his secret section π_s :

$$\pi_s(\delta) = \pi_s(\varphi(\beta)) * \pi_s(\tilde{\tau}) = \varphi(\pi_s(\beta)) * \tau$$

Note that the fact that π_s is a section says exactly that for any $a \in A$, $\pi_s(\tilde{a}) = a$, where \tilde{a} is the

image of a under the defining A -algebra map $A \rightarrow A_{\varphi,v}$. Now, Bob computes the symbol:

$$\text{symb}_{\varphi}(\pi_s(\delta)) = \text{symb}_{\varphi}(\varphi(\pi_s(\beta))) * \text{symb}_{\varphi}(\tau) = \text{symb}_{\varphi}(\tau) = \kappa$$

The key fact that allows the decryption to work is that $\pi_s(\beta) \in G(A)$ rather than $G(A_{\varphi,v})$, so $\varphi(\pi_s(\beta)) \in \varphi(G(A)) \subseteq \ker(\text{symb}_{\varphi})$.

Definition 3.7.1. To decrypt a message $C = (C_1, \dots, C_{\ell(n_A-1)})$ using his secret key s , Bob runs:

$$\begin{aligned} v &= \varphi(s); \\ (c, \delta) &= C_i \text{ where } v = \text{pk}_B * \alpha_i; \\ y &= \pi_s(\delta); \\ \kappa &= \text{symb}_{\varphi}(y); \\ \text{DECRYPT}(C, \delta, s) &= \kappa^{-1} \cdot c; \end{aligned}$$

The discussion above proves:

Theorem 3.7.2 (Soundness of decryption). *For any message $m \in C^L$ and any identifier id_B with corresponding secret key $s = \text{sk}_B$, we have*

$$\text{DECRYPT}(\text{ENCRYPT}(m, \text{id}_B), s) = m$$

3.8 Security

Next, we discuss the security of this encryption scheme and show that it reduces to Problem 3.4.1. The main fact we will exploit is that for $w \in \ker(\text{symb}_{\varphi}) - \varphi(G(A))$, the result of running $\mathcal{E}(m, w)$ contains *literally* no information about m , in a strong information-theoretic sense:

Lemma 3.8.1. *Suppose that $w \in \ker(\text{symb}_{\varphi}) - \varphi(G(A))$. Then the random variable $\mathcal{E}(m, w)$, distributed over the random choices of $\tau \in H(A)^L$ and $\alpha \in H(A_{\varphi,w})^L$ made in Definition 3.6.1, is statistically independent from $m \in C^L$: for any two $m, m' \in \mathcal{M} = C^L$, the distributions of $\mathcal{E}(m, w)$ and $\mathcal{E}(m', w)$ are identical.⁴*

We note that this lemma can be phrased as stating that when $w \notin \varphi(G(A))$, $\mathcal{E}(m, w)$ is a *perfectly secure Shannon cipher* of the message m : see Section 2.2 for some context for the statistical results in this section. This means that it is impossible for an eavesdropper to deduce any information about m from $\mathcal{E}(m, w)$, even if the eavesdropper is allowed to have no bounds on their computational efficiency.

Thus, if an eavesdropper is able to perform some computation on the total ciphertext

$$(\mathcal{E}(m, \text{pk}_{\text{id}} * \alpha_i))_{i=1}^{\ell(n_A-1)}$$

⁴The length of the message is always considered as a common-knowledge parameter: the goal of encryption is to obfuscate the message *among all messages of the same length*

which enables them to learn any information about m (as made precise by the notion of IND-ID-CPA security, as in Definition 2.5.1), they must be able to somehow distinguish the component $\mathcal{E}(m, \varphi(\text{sk}_{\text{id}}))$ from the “random noise” contained in the other components. However, if the isogeny image assumption of Problem 3.4.1 is true, then it is impossible for any efficient adversary to distinguish elements of $\varphi(G(A))$ among $\ker(\text{symb}_\varphi)$, implying it is impossible to distinguish the output distribution of $\mathcal{E}(m, \varphi(\text{sk}_{\text{id}}))$ from that of the other components. In particular, it is impossible to detect that $\mathcal{E}(m, \varphi(\text{sk}_{\text{id}}))$ is not also independent from m ! We’ll make this deduction precise in Theorem 3.8.2.

Proof. First, let us examine the construction of $\mathcal{E}(m, w)$ more closely when $w \in \ker(\text{symb}_\varphi)$ but $w \notin \varphi(G(A))$.

As above, we define $A_{\varphi, w}$ by

$$\text{Spec}(A_{\varphi, w}) = \varphi^{-1}(w) = G \times_{H, \varphi, w} \text{Spec}(A)$$

This a \underline{C} -torsor over $\text{Spec}(A)$: in particular, $A_{\varphi, w}$ is a finite étale A -algebra of constant rank ℓ .

Since $w \in \ker(\text{symb}_\varphi)$ but not in $\varphi(G(A))$, there must be at least two prime ideals $\mathfrak{r} \in \text{Spec}(A)$ for which $\text{symb}_\mathfrak{r}(w) \neq 1$. Let the set of such prime ideals be $\mathcal{R} = \{\mathfrak{r}_1, \dots, \mathfrak{r}_j\}$. Since the local symbols detect the image of $\varphi(G(\mathbf{F}_\mathfrak{r}))$ exactly, \mathcal{R} is also the set of prime ideals such that $w_\mathfrak{r} \notin \varphi(G(\mathbf{F}_\mathfrak{r}))$.

We can write $A = A_0 \times A_1$ with $|\text{Spec}(A_0)| = \mathcal{R}$ and $|\text{Spec}(A_1)| = |\text{Spec}(A)| - \mathcal{R}$, and we can correspondingly decompose $A_{\varphi, w} = A_{0, \varphi, w} \times A_{1, \varphi, w}$. (Allowing the possibility that $A_1 = 0$). Then we will have $w = (w_0, w_1) \in H(A) = H(A_0) \times H(A_1)$ with $w_1 \in \varphi(G(A_1))$. In particular, we will have $A_{1, \varphi, w} \simeq \prod_{i=1}^\ell A_1$, as in (3.6.3). Above each of the ideals $\mathfrak{r} \in \mathcal{R}$, the fiber of $A_{\varphi, w}$ is a non-split étale algebra of degree ℓ over $\mathbf{F}_\mathfrak{r}$: as ℓ is *prime*, this is a degree- ℓ *field extension* of $\mathbf{F}_\mathfrak{r}$ (see Proposition 3.1.1).

Looking back at Definition 3.6.1, we have $\mathcal{E}(m, w) = (c, \delta) = (\kappa \cdot m, \delta)$, where $\kappa = \text{symb}_\varphi(\tau)$ for a randomly chosen $\tau \in H(A)^L$. As symb_φ is a group homomorphism, it follows that κ is a uniformly random element of C^L . Therefore, for any fixed $m \in C^L$, $c = \kappa \cdot m$ is also a uniformly distributed element of C^L . In particular, for any two messages $m, m' \in \mathcal{M} = C^L$, the distributions of the random variables $\kappa \cdot m$ and $\kappa \cdot m'$ are the same. In the sense of Section 2.2, as a randomized function of m , $c = \kappa \cdot m$ is a perfectly secure Shannon cipher. (This should not be suprising; it is precisely a one-time pad.)

Our goal is to prove that this property of independence from m still holds for the full $\mathcal{E}(m, w) = (c, \delta)$ rather than just c . Recall that δ is defined by sampling β from $G(A_{\varphi, w})^L$ uniformly at random, then setting

$$\delta = \varphi(\beta) * \tilde{\tau} \tag{3.8.1}$$

inside of $H(A_{\varphi, w})$. Loosely, the reason for this independence will come from the fact that the local symbol of τ at primes in \mathcal{R} is no longer encoded in δ . The fiber of $A_{\varphi, w}$ over such primes is a non-trivial extension, expanding the image of φ , so multiplication by $\varphi(\beta)$ no longer preserves the

symbol. In fact, it will be the case that when $\mathfrak{r} \in \mathcal{R}$, for any $\delta_{\mathfrak{r}}, \tau_{\mathfrak{r}}$, it will always be possible to solve the \mathfrak{r} -factor of (3.8.1) for $\beta_{\mathfrak{r}}$. Then, as

$$\kappa = \text{symb}_{\varphi}(\tau) = \prod_{\mathfrak{r} \in |\text{Spec}(A)|} \text{symb}_{\mathfrak{r}}(\tau_{\mathfrak{r}}), \quad (3.8.2)$$

despite the fact that $\delta_{\mathfrak{r}}$ still encodes the symbol of $\tau_{\mathfrak{r}}$ for $\mathfrak{r} \notin \mathcal{R}$, fixing δ puts no constraint on the product κ of the symbols. In other words, for any δ, κ , we will be able to solve (3.8.1) and (3.8.2) for τ, β . As everything in sight is a group homomorphism, the number of such solutions (τ, β) will then be constant (it will be equal to $\#\ker(\text{symb}_{\varphi}) * \#(\ker \varphi)(A_{\varphi, w})$); as τ and β are both uniformly random, this implies that any⁵ δ, κ is equally likely. Therefore, observing δ tells us nothing about κ , and therefore that observing $(c, \delta) = (\kappa m, \delta)$ tells us nothing about m .

Now, we will fill in the details in this argument by making a careful analysis of the distribution of δ . To do this, we will take a look at the distributions of each of $\varphi(\beta)$ and $\tilde{\tau}$, working over one prime ideal \mathfrak{r} of A at a time. There are two cases: either $\mathfrak{r} \in \mathcal{R}$ or $\mathfrak{r} \notin \mathcal{R}$.

Case 1: ($\mathfrak{r} \in \mathcal{R}$) The primes in \mathcal{R} are those for which $v_{\mathfrak{r}}$ is not in the image $\varphi(G(\mathbf{F}_{\mathfrak{r}}))$, and thus the extension of $\mathbf{F}_{\mathfrak{r}}$ obtained by adjoining a preimage of $v_{\mathfrak{r}}$ under φ is non-trivial. The effect of this is that the image of $\varphi(G(A_{\varphi, w} \otimes_A \mathbf{F}_{\mathfrak{r}}))$ is larger than $\varphi(G(\mathbf{F}_{\mathfrak{r}}))$. In particular, it contains $\tilde{v}_{\mathfrak{r}}$ (where for $g \in G(A)$ we write \tilde{g} for the image of g under the “restriction” map $G(A) \rightarrow G(A_{\varphi, w})$). This will make it impossible to learn anything about the symbol $\text{symb}_{\mathfrak{r}}(\tau_{\mathfrak{r}})$ from δ , in contrast to the split case where we were able to recover this symbol using Bob’s private key. (See Section 3.7).

First, we study β . Taking components at each prime ideal $\mathfrak{r} \in |\text{Spec}(A)|$, we can write

$$\beta = (\beta_{\mathfrak{r}})_{\mathfrak{r} \in |\text{Spec}(A)|} \in G(A_{\varphi, w}) = \prod_{\mathfrak{r} \in |\text{Spec}(A)|} G(A_{\varphi, w} \otimes_A \mathbf{F}_{\mathfrak{r}})$$

By definition, β is a uniformly distributed random variable on this finite product space. This is equivalent to the following two statements:

- (a) For each $\mathfrak{r} \in |\text{Spec}(A)|$, $\beta_{\mathfrak{r}}$ is a uniformly distributed random variable on $G(A_{\varphi, w} \otimes_A \mathbf{F}_{\mathfrak{r}})$
- (b) These n_A random variables $\{\beta_{\mathfrak{r}}\}_{\mathfrak{r} \in |\text{Spec}(A)|}$ are independently distributed with respect to one another.

In particular, the same is true if we restrict attention to the primes $\mathfrak{r} \in \mathcal{R}$. For these primes, we abuse notation a little and write

$$\mathbf{F}_{\mathfrak{r}^{\ell}} := A_{\varphi, w} \otimes_A \mathbf{F}_{\mathfrak{r}}$$

⁵We will need to restrict the domain of possible δ slightly for this statement to be true, but this restriction only depends on w , so the conclusion is still true: see what follows.

Note that since $\mathfrak{r} \in \mathcal{R}$ and ℓ is prime, $\mathbf{F}_{\mathfrak{r}^\ell}$ is a field extension of $\mathbf{F}_{\mathfrak{r}}$ of degree ℓ : there is only one such field extension up to isomorphism, so the abuse of notation is not unreasonable.

Now, we have a collection of $\#(\mathcal{R})$ independent random variables $\beta_{\mathfrak{r}}$ which are uniformly distributed on $G(\mathbf{F}_{\mathfrak{r}^\ell})$. Since φ is a group homomorphism from the finite group $G(A_{\varphi,w})^L$ onto the finite group $\varphi(G(A_{\varphi,w}))^L \subseteq H(A_{\varphi,v})^L$ (and φ respects the product decomposition according to the prime ideals of A), we see that the $\varphi(\beta_{\mathfrak{r}})_{\mathfrak{r} \in \mathcal{R}}$ are a collection of independent uniformly distributed random variables on the images $\varphi(G(\mathbf{F}_{\mathfrak{r}^\ell}))$.

Now, we recall the key computations we used in our definition of the local symbol in Section 3.2. First, we had the exact sequence (3.2.3):

$$0 \rightarrow C \rightarrow G(\mathbf{F}) \rightarrow H(\mathbf{F}) \rightarrow H^1(\mathbf{F}, C) \rightarrow 0$$

Then, we applied the identification (3.2.4)

$$H^1(\mathbf{F}, C) = \text{Hom}(\Gamma_{\mathbf{F}}, C) \xrightarrow[F]{\sim} C$$

Both of these are valid for any finite field, and together, they give the local symbol $\text{symb}_{\mathbf{F}}: H(\mathbf{F}) \rightarrow C$ whose kernel is exactly $\varphi(G(\mathbf{F}))$. Thus, we may restate our conclusions above as saying that the $\varphi(\beta_{\mathfrak{r}})_{\mathfrak{r} \in \mathcal{R}}$ form a collection of independent random variables over the sets $\{\ker(\text{symb}_{\mathbf{F}_{\mathfrak{r}^\ell}})\}_{\mathfrak{r} \in \mathcal{R}}$.

Next, we can apply the pullback functoriality of Galois (or étale) cohomology to obtain the following commutative diagram:

$$\begin{array}{ccccccccc}
 0 & \longrightarrow & C & \longrightarrow & G(\mathbf{F}_{\mathfrak{r}}) & \xrightarrow{\varphi} & H(\mathbf{F}_{\mathfrak{r}}) & \xrightarrow{\text{symb}_{\mathfrak{r}}} & \text{Hom}(\Gamma_{\mathfrak{r}}, C) & \xrightarrow{\sim} & C & \longrightarrow & 0 \\
 & & \downarrow \text{id} & & \downarrow & & \downarrow \tau \mapsto \tilde{\tau} & & \downarrow \text{res}=0 & & \downarrow (\cdot)^\ell=0 & & \\
 0 & \longrightarrow & C & \longrightarrow & G(\mathbf{F}_{\mathfrak{r}^\ell}) & \xrightarrow{\varphi} & H(\mathbf{F}_{\mathfrak{r}^\ell}) & \xrightarrow{\text{symb}_{\mathfrak{r}^\ell}} & \text{Hom}(\Gamma_{\mathfrak{r}^\ell}, C) & \xrightarrow{\sim} & C & \longrightarrow & 0
 \end{array} \tag{3.8.3}$$

Note that the Galois group $\Gamma_{\mathfrak{r}^\ell}$ is the subgroup of $\Gamma_{\mathfrak{r}}$ generated by $(\text{Frob}_{\mathfrak{r}})^\ell$, so the right-most square of the diagram commutes. As C is ℓ -torsion and the horizontal arrows of this square are isomorphisms, we conclude that the restriction map from $\text{Hom}(\Gamma_{\mathfrak{r}}, C)$ to $\text{Hom}(\Gamma_{\mathfrak{r}^\ell}, C)$ is 0. Chasing this diagram, we see that for any $\tau \in H(\mathbf{F}_{\mathfrak{r}})^L$, $\tilde{\tau}$ is inside of $\ker(\text{symb}_{\mathbf{F}_{\mathfrak{r}^\ell}})$, and therefore $\tilde{\tau} \in \varphi(G(\mathbf{F}_{\mathfrak{r}^\ell}))$.

Now, we return to the definition of δ :

$$\delta = \varphi(\beta) * \tilde{\tau} \in H(A_{\varphi,w})^L$$

Taking components at each prime ideal $\mathfrak{r} \in \mathcal{R}$, we have:

$$\delta_{\mathfrak{r}} = \varphi(\beta_{\mathfrak{r}}) * \tilde{\tau}_{\mathfrak{r}}$$

We saw above that the $\{\varphi(\beta_{\mathfrak{r}})\}$ comprise a collection of independent and uniformly distributed random variables over the images $\varphi(G(\mathbf{F}_{\mathfrak{r}^\ell}))$; as the $\tilde{\tau}_{\mathfrak{r}}$ are also in these images, we see that for any fixed choice of τ , the $\delta_{\mathfrak{r}}$ are also uniformly distributed in the $\varphi(G(\mathbf{F}_{\mathfrak{r}^\ell}))$, so we've shown:

Proposition. *Let $w \in \ker(\text{symb}_\varphi)$ and \mathfrak{r} be such that $w_{\mathfrak{r}} \notin \varphi(G(\mathbf{F}_{\mathfrak{r}}))$. Then, given any $\tau_{\mathfrak{r}} \in H(\mathbf{F}_{\mathfrak{r}})^L$, the conditional distribution $P(\delta_{\mathfrak{r}}|\tau_{\mathfrak{r}})$ of the random variable $\delta_{\mathfrak{r}}$ is uniform on the image $\varphi(G(A_{\varphi,w} \otimes_A \mathbf{F}_{\mathfrak{r}}))$. In particular, for such \mathfrak{r} , the random variables $\tau_{\mathfrak{r}}$ and $\delta_{\mathfrak{r}}$ are independent.*

Case 2: ($\mathfrak{r} \notin \mathcal{R}$)

The fiber of $A_{\varphi,w}$ over primes $\mathfrak{r} \in |\text{Spec}(A)| - \mathcal{R}$ is isomorphic as $\mathbf{F}_{\mathfrak{r}}$ -algebra to $\prod_{j=1}^{\ell} \mathbf{F}_{\mathfrak{r}}$ with the $\mathbf{F}_{\mathfrak{r}}$ -algebra structure given by the diagonal embedding, similarly to (3.6.3) (see Proposition 3.1.1). In particular, given $\tau_{\mathfrak{r}} \in H(\mathbf{F}_{\mathfrak{r}})$, the “restriction” $\tilde{\tau}_{\mathfrak{r}}$ is the diagonal image $(\tau_{\mathfrak{r}})_{j=1}^{\ell}$. We'll sometimes write $(\mathbf{F}_{\mathfrak{r}})^{\ell}$ for this product algebra.

Now, $\beta_{\mathfrak{r}}$ is a uniformly distributed random variable on

$$G(A_{\varphi,w} \otimes_A \mathbf{F}_{\mathfrak{r}}) \simeq \prod_{j=1}^{\ell} G(\mathbf{F}_{\mathfrak{r}}),$$

so we have $\beta_{\mathfrak{r}} = (\beta_{\mathfrak{r},j})_{j=1}^{\ell}$, and the $\{\beta_{\mathfrak{r},j}\}$ are a collection of ℓ independent and uniformly distributed random variables on $G(\mathbf{F}_{\mathfrak{r}})$. Applying the group homomorphism φ , we see that the $\{\varphi(\beta_{\mathfrak{r},j})\}$ are a collection of ℓ independent and uniformly distributed random variables on $\varphi(G(\mathbf{F}_{\mathfrak{r}})) = \ker(\text{symb}_{\mathfrak{r}})$. Fixing some $\tau_{\mathfrak{r}} \in H(\mathbf{F}_{\mathfrak{r}})^L$, and multiplying $\varphi(\beta_{\mathfrak{r}})$ by $\tilde{\tau}_{\mathfrak{r}} = (\tau_{\mathfrak{r}})_{j=1}^{\ell}$, we see:

$$\delta_{\mathfrak{r}} = (\delta_{\mathfrak{r},j})_{j=1}^{\ell} = (\tau_{\mathfrak{r}} * \varphi(\beta_{\mathfrak{r},j}))_{j=1}^{\ell}$$

By the above we see that the conditional distribution of $\delta_{\mathfrak{r}}$ given $\tau_{\mathfrak{r}}$ is uniform on the set

$$H((\mathbf{F}_{\mathfrak{r}})^{\ell})_{\text{diag},\sigma}^L := \left\{ (h_j)_{j=1}^{\ell} \in \prod_{j=1}^{\ell} H(\mathbf{F}_{\mathfrak{r}})^L \mid \text{for all } j = 1, \dots, \ell, \text{symb}_{\mathfrak{r}}(h_j) = \sigma \right\}, \quad (3.8.4)$$

where $\sigma := \text{symb}_{\mathfrak{r}}(\tau_{\mathfrak{r}})$.

In particular, note that no matter what the value of $\tau \in H(A)^L$ is, we see that $\delta_{\mathfrak{r}}$ is restricted

to the following set:

$$\delta_{\mathfrak{r}} \in H((\mathbf{F}_{\mathfrak{r}})^{\ell})_{\text{diag}}^L := \bigcup_{\sigma \in C^L} H((\mathbf{F}_{\mathfrak{r}})^{\ell})_{\text{diag}, \sigma}^L \quad (3.8.5)$$

$$= \left\{ (h_j)_{j=1}^{\ell} \in \prod_{j=1}^{\ell} H(\mathbf{F}_{\mathfrak{r}})^L \mid \text{for all } j \neq j', \text{ symb}_{\mathfrak{r}}(h_j) = \text{symb}_{\mathfrak{r}}(h_{j'}) \right\} \quad (3.8.6)$$

Now, we've shown:

Proposition. *Let $w \in \ker(\text{symb}_{\varphi})$ and \mathfrak{r} be such that $w_{\mathfrak{r}} \in \varphi(G(\mathbf{F}_{\mathfrak{r}}))$. Fix some $\tau_{\mathfrak{r}} \in H(\mathbf{F}_{\mathfrak{r}})^L$ and set $\sigma = \text{symb}_{\mathfrak{r}}(\tau_{\mathfrak{r}})$. Then the conditional distribution $P(\delta_{\mathfrak{r}} | \tau_{\mathfrak{r}})$ of the random variable $\delta_{\mathfrak{r}}$ is uniform on the set $H((\mathbf{F}_{\mathfrak{r}})^{\ell})_{\text{diag}, \sigma}$*

Putting the two cases together, we get:

Proposition. *Let $w \in \ker(\text{symb}_{\varphi})$, and let \mathcal{R} be the set of prime ideals of A such that $\text{symb}_{\mathfrak{r}}(\mathbf{F}_{\mathfrak{r}}) \neq 1$. Fix some $\tau \in H(\mathbf{F}_{\mathfrak{r}})^L$, and for $\mathfrak{r} \in |\text{Spec}(A)| - \mathcal{R}$, set $\sigma_{\mathfrak{r}} := \text{symb}_{\mathfrak{r}}(\tau_{\mathfrak{r}})$, $\sigma = (\sigma_{\mathfrak{r}})_{\mathfrak{r} \in |\text{Spec}(A)| - \mathcal{R}}$. Then the conditional distribution of δ given τ is the uniform distribution on the set*

$$\mathcal{D}(\mathcal{R}; \sigma) := \prod_{\mathfrak{r} \in \mathcal{R}} \ker(\text{symb}_{\mathbf{F}_{\mathfrak{r}^{\ell}}})^L \times \prod_{\mathfrak{r} \in |\text{Spec}(A)| - \mathcal{R}} H(\mathbf{F}_{\mathfrak{r}})_{\text{diag}, \sigma_{\mathfrak{r}}}^L \quad (3.8.7)$$

$$\subseteq H(A_{\varphi, w})^L \simeq \prod_{\mathfrak{r} \in \mathcal{R}} H(\mathbf{F}_{\mathfrak{r}^{\ell}})^L \times \prod_{\mathfrak{r} \in |\text{Spec}(A)| - \mathcal{R}} H((\mathbf{F}_{\mathfrak{r}})^{\ell})^L \quad (3.8.8)$$

The overall (“marginal”) distribution of δ is uniform on the set

$$\mathcal{D}(\mathcal{R}) := \prod_{\mathfrak{r} \in \mathcal{R}} \ker(\text{symb}_{\mathbf{F}_{\mathfrak{r}^{\ell}}})^L \times \prod_{\mathfrak{r} \in |\text{Spec}(A)| - \mathcal{R}} H(\mathbf{F}_{\mathfrak{r}})_{\text{diag}}^L \quad (3.8.9)$$

Note that the last statement of the Proposition follows from the first by taking the average over τ ; as τ is uniformly distributed, $\sigma_{\mathfrak{r}}$ is uniformly distributed in C^L for each \mathfrak{r} .

Now, recall that our goal is to show, for any fixed $w \in \ker(\text{symb}_{\varphi}) - \varphi(G(A))$, that the distribution of the random variable $\mathcal{E}(m, w) = (\kappa \cdot m, \delta)$ over the space of uniformly random choices of $\tau \in H(A)^L$, $\beta \in H(A_{\varphi, w})^L$ does not depend on m .

It suffices to show that the random variables κ and δ are independent from each other: $\kappa \cdot m$ is uniformly distributed in C^L independently of m , so if δ - whose definition does not involve m at all - is distributed independently of κ , it follows that the full $\mathcal{E}(m, w)$ has distribution independent from m .

Intuitively, what's going on here is that $\kappa = \text{symb}_{\varphi}(\tau) = \prod_{\mathfrak{r} \in |\text{Spec}(A)|} \text{symb}_{\mathfrak{r}}(\tau_{\mathfrak{r}})$. However, we saw above, that for the primes $\mathfrak{r} \in \mathcal{R}$, $\delta_{\mathfrak{r}}$ and $\tau_{\mathfrak{r}}$ are independent. Thus, even fixing δ , there are at

least two terms in that product which are allowed to vary independently, allowing any possibility for the value of κ .

To make this idea rigorous, we can perform the following computation of the joint probability $P(\kappa, \delta)$. Fix an arbitrary $\kappa \in C^L$. Considering $\delta \in H(A_{\varphi, w})^L$, we saw above that $P(\delta) = 0$ unless

$$\delta \in \mathcal{D}(\mathcal{R}) = \bigcup_{\sigma} \mathcal{D}(\mathcal{R}; \sigma)$$

where σ ranges over $\prod_{\mathfrak{r} \in |\text{Spec}(A)| - \mathcal{R}} C^L$. Now, fix an arbitrary $\delta \in \mathcal{D}(\mathcal{R})$, and let σ be the unique element of $\prod_{\mathfrak{r} \in |\text{Spec}(A)| - \mathcal{R}} C^L$ such that $\delta \in \mathcal{D}(\mathcal{R}; \sigma)$.

Then:

$$P(\kappa, \delta) = \sum_{\tau \in H(A)^L} P(\kappa, \delta \mid \tau) P(\tau) \quad (3.8.10)$$

$$= \frac{1}{\#H(A)^L} \sum_{\tau \in H(A)^L} P(\kappa, \delta \mid \tau) \quad (3.8.11)$$

$$= \frac{1}{\#H(A)^L} \sum_{\substack{\tau \in H(A)^L \\ \text{symb}_{\varphi}(\tau) = \kappa}} P(\delta \mid \tau) \quad (3.8.12)$$

$$= \frac{1}{\#H(A)^L} \sum_{\substack{\tau \in H(A)^L \\ \text{symb}_{\varphi}(\tau) = \kappa \\ \text{symb}_{\mathcal{R}}(\tau) = \sigma}} \frac{1}{\#\mathcal{D}(\mathcal{R}; \sigma)} \quad (3.8.13)$$

$$= \frac{\#\{\tau \in H(A)^L : \text{symb}_{\varphi}(\tau) = \kappa, \text{symb}_{\mathcal{R}}(\tau) = \sigma\}}{\#H(A)^L \cdot \#\mathcal{D}(\mathcal{R}; \sigma)} \quad (3.8.14)$$

In (3.8.13) and (3.8.14), the notation $\text{symb}^{\mathcal{R}}(\tau)$ means $(\text{symb}_{\mathfrak{r}}(\tau))_{\mathfrak{r} \in |\text{Spec}(A)| - \mathcal{R}}$: in words, we sum over $\tau \in H(A)^L$ such that $\text{symb}_{\varphi}(\tau) = \kappa$ and $\text{symb}_{\mathfrak{r}}(\tau_{\mathfrak{r}}) = \sigma_{\mathfrak{r}}$ for all $\mathfrak{r} \in |\text{Spec}(A)| - \mathcal{R}$.

Let's break down how this computation works. First, we get (3.8.10) from the law of total expectation. Next, we get (3.8.11) by noting that τ is - by construction - uniformly distributed on $H(A)^L$. Then, to get (3.8.12), note that κ is a deterministic function of τ : $\kappa = \text{symb}_{\varphi}(\tau)$. Thus, the probability of (κ, δ) given τ is 0 unless $\kappa = \text{symb}_{\varphi}(\tau)$; when this equation holds, the joint probability of κ, δ given τ is just the probability of δ given τ . Next, we apply our above analysis of the distribution of δ given τ to derive (3.8.13): given τ , δ is uniformly distributed on $\mathcal{D}(\mathcal{R}; \text{symb}^{\mathcal{R}}(\tau))$. Since we fixed a $\delta \in \mathcal{D}(\mathcal{R}; \sigma)$, the terms of (3.8.12) with $\text{symb}^{\mathcal{R}}(\tau) \neq \sigma$ are 0. Finally, to get (3.8.14), we note that the sum in (3.8.13) is constant.

We now argue that the value of (3.8.14) does not depend on $\delta \in \mathcal{D}(\mathcal{R})$ or $\kappa \in C^L$; in other words, we prove that the distribution of (κ, δ) is uniform on the space $C^L \times \mathcal{D}(\mathcal{R})$. Multiplying κ by some fixed $m \in C^L$ does not change this distribution, so we get an even stronger statement than that of the Lemma: not only does the distribution of $\mathcal{E}(m, w) = (\kappa \cdot m, \delta)$ not depend on m , but in fact it

is the uniform distribution on a set with a simple and uniform distribution depending only on the set \mathcal{R} .

First of all, we note that the size of the set $\mathcal{D}(\mathcal{R}; \sigma)$ does not depend on σ ; this follows immediately, for example by examining our derivation of (3.8.7) and noting that we constructed $\mathcal{D}(\mathcal{R}; \sigma)$ as a coset for the group

$$\varphi(G(A_{\varphi,w})) = \prod_{\mathbf{r} \in \mathcal{R}} \ker(\text{symb}_{\mathbf{F}_{\mathbf{r}^\ell}}) \times \prod_{\mathbf{r} \in |\text{Spec}(A)| - \mathcal{R}} \prod_{j=1}^{\ell} \ker(\text{symb}_{\mathbf{r}})$$

inside of $H(A_{\varphi,w})$.

Now, all that remains is to show that the count appearing in the numerator of (3.8.14) does not depend on σ or κ . This is just a little bit of linear algebra (and another place where the assumption that ℓ is prime is very helpful). Indeed, as $\#(\mathcal{R}) \geq 2$, it is always possible to solve the following $n_A - \#(\mathcal{R})$ simultaneous equations in the n_A variables $\{\text{symb}_{\mathbf{r}}(\tau_{\mathbf{r}})\}_{\mathbf{r} \in |\text{Spec}(A)|}$

$$\text{symb}_{\varphi}(\tau) = \prod_{\mathbf{r} \in \text{Spec}(A)} \text{symb}_{\mathbf{r}}(\tau_{\mathbf{r}}) = \kappa \quad (3.8.15)$$

$$\forall \mathbf{r} \in |\text{Spec}(A)| - \mathcal{R}: \quad \text{symb}_{\mathbf{r}}(\tau_{\mathbf{r}}) = \sigma_{\mathbf{r}} \quad (3.8.16)$$

Indeed, there will always be $\ell^{\#(\mathcal{R})-1}$ values of $(\text{symb}_{\mathbf{r}}(\tau_{\mathbf{r}}))_{\mathbf{r} \in |\text{Spec}(A)|}$ that satisfy these equations; then, since the collection $(\text{symb}_{\mathbf{r}}(\tau_{\mathbf{r}}))_{\mathbf{r} \in |\text{Spec}(A)|}$ uniquely determines $\tau \in H(A)^L$ modulo $\varphi(G(A))^L$, there are $\#(\varphi(G(A))^L)$ values of τ that solve these equations. In particular, this count depends on neither δ nor κ , proving our desired independence statement! \square

The above observations allow us to prove our main security result:

Theorem 3.8.2. *For any choice of isogeny $\varphi: G \rightarrow H$ as above, the corresponding IBE scheme is IND-ID-CPA secure in the random oracle model under the isogeny image assumption of Definition 3.4.1 for φ .*

Proof. This proof is loosely adapted from the proof of security of the Cocks IBE scheme in [17].

Suppose we are given an adversary \mathcal{A} for the IND-ID-CPA game as described in Definition 2.5.1 which is able to achieve a non-negligible advantage. We use this to construct a *simulator* algorithm \mathcal{S} which obtains non-negligible advantage in the isogeny image problem game of Problem 3.4.1 by interacting with \mathcal{A} .

Define \mathcal{S} to have as input the public parameters **params** and an element $h \in H(A)$ with $\text{symb}_{\varphi}(h) = 1$. Its task is to determine whether $h \in \varphi(G(A))$ by interacting with the IND-ID-CPA adversary \mathcal{A} . The algorithm \mathcal{S} will play the role of challenger in the IND-ID-CPA game as well as the role of the random oracle.

First, it passes **params** to \mathcal{A} . Then \mathcal{A} proceeds as in the IND-ID-CPA game. Suppose that over the course of running \mathcal{A} makes queries on a total of Q distinct identities $\text{id}_1, \dots, \text{id}_Q$. These can either be queries to the random oracle or secret-key identity queries asked of the challenger in the IND-ID-CPA game. We assume without loss of generality that they are all secret-key identity queries (i.e. even if \mathcal{A} only asks for the public key associated by the random oracle to an identity id as $\mathcal{H}(\text{id})$, we include id on the list of secret key queries \mathcal{A} is allowed to ask: the secret key provides strictly more information than the public key).

To respond to the queries from \mathcal{A} , we keep a running list of triples (id, s, α_i) with $s \in G(A)$ and $\alpha_i \in H(A)$ one of those appearing in **params**. If \mathcal{A} makes a query on some id , we check it against this list: if the query is for an identity that has already appeared, we respond with either s or $\alpha_i \varphi(s)$ according to whether the query is a secret key query or a random oracle query respectively. Otherwise, we choose some $s \in G(A)$ and $\alpha_i \in \{\alpha_1, \dots, \alpha_\ell\}$ uniformly at random and add (id, s, α_i) to our table, then respond with either s or $\alpha_i * \varphi(s)$ according to whether the query is a secret key query or a random oracle query.

From the perspective of \mathcal{A} , this is no different from interacting with a genuine random oracle and a challenger in the IND-ID-CPA game possessing the master secret key: in both cases, the public key associated to an identity is a uniformly random sample from the set of elements $h \in H(A)$ with $\text{symb}_\varphi(h) = 1$, as such elements are uniformly distributed among the cosets $\{\alpha_i * \varphi(G(A))\}$.

When \mathcal{A} is ready to choose a “challenge identity” id^* , we send it h when it queries the random oracle to get $\text{pk}^* = \mathcal{H}(\text{id}^*)$. It responds with a pair of messages m_0, m_1 such that \mathcal{A} is able to distinguish between the encryption of m_0 and m_1 for id^* with non-negligibly better than random accuracy.

We respond to \mathcal{A} by choosing a random bit b and, rather than sending \mathcal{A} the ciphertext

$$\text{ENCRYPT}(m_b, \text{id}^*) = (\mathcal{E}(m, \alpha_i * \text{pk}^*))_{i=1}^{\ell * (n_A - 1)} \quad (3.8.17)$$

from Definition 3.6.1, we send \mathcal{A}

$$(\mathcal{E}(m, h) = \mathcal{E}(m, \alpha_1 * \text{pk}^*), \xi_2, \dots, \xi_{\ell * (n_A - 1)}) \quad (3.8.18)$$

with $(\xi_2, \dots, \xi_{\ell * (n_A - 1)})$ drawn uniformly at random from the ciphertext space C^L . Then, if \mathcal{A} guesses correctly, we output 1 - indicating that h is in the image of φ - and otherwise we output 0. The idea here is that by replacing the other components of the ciphertext with noise, we are forcing \mathcal{A} to base its guess on information derived from $\mathcal{E}(m, h)$. But Lemma 3.8.1 implies that the only case in which $\mathcal{E}(m, h)$ contains any information about m is when $h \in \varphi(G(A))$, so \mathcal{A} is more likely to be successful when this is the case.

Now, we compute the probability that our response is correct. There are two cases:

- $h \in \varphi(G(A))$: this case occurs with probability $\frac{1}{\ell * (n_A - 1)} = \#(\ker(\text{symb}_\varphi) / \varphi(G(A)))^{-1}$. As

$h = \text{pk}^* = \alpha_1 * \text{pk}^*$, for $i > 1$, $\alpha_i * \text{pk}^*$ is not in $\varphi(G(A))$. Thus, Lemma 3.8.1 says that for $i > 1$, the distribution of $\mathcal{E}(m_b, \alpha_i * \text{pk}^*)$ is identical to that of ξ_i (i.e. uniform on C^L). Since our response (3.8.18) to \mathcal{A} includes the actual value of $\mathcal{E}(m_b, \alpha_1 * \text{pk}^*) = \mathcal{E}(m_b, h)$ in its α_1 -component, its distribution is identical to that of the actual ciphertext (3.8.17).

Thus, by assumption, \mathcal{A} can guess b from this response with the some non-negligible advantage ϵ : this means that the conditional probability that \mathcal{A} is correct given that $h \in \varphi(G(A))$ is $\frac{1}{2} + \epsilon$. Our output is thus 1 with conditional probability $\frac{1}{2} + \epsilon$ and 0 with conditional probability $\frac{1}{2} - \epsilon$: as the correct output in this case is 1, we are successful with conditional probability $\frac{1}{2} + \epsilon$ in this case.

- h is not in $\varphi(G(A))$: By applying Lemma 3.8.1 again, we see that the distribution of $\mathcal{E}(m, h)$ is independent from m . Since the other components of our response (3.8.18) are drawn uniformly at random from C^L , our entire response is statistically independent from m , so \mathcal{A} has no hope of guessing b with non-negligible advantage. Therefore, \mathcal{A} is equally likely to be correct as incorrect, since our choice of random bit is fully independent from the message we send to \mathcal{A} . In this case, we output 1 with probability $\frac{1}{2}$ and 0 with probability $\frac{1}{2}$; as the correct answer in this case is 0, we are successful with probability $\frac{1}{2}$.

Altogether, our success probability is

$$\frac{1}{\ell * (n_A - 1)} \cdot \left(\frac{1}{2} + \epsilon \right) + \frac{\ell * (n_A - 1) - 1}{\ell * (n_A - 1)} \cdot \frac{1}{2} = \frac{1}{2} + \frac{\epsilon}{\ell * (n_A - 1)}$$

As ϵ is non-negligible, and ℓ and n_A are both constant with respect to λ , our success probability is also non-negligible, so we break the isogeny image assumption. \square

Chapter 4

Cocks-Kummer IBE for number fields

Now, we will specialize our construction to a setting where we can derive an efficient algorithm for computing the symbol symb_φ *without* knowing the factorization $N = pq$ of the master key. The technique to do so will use *class field theory*, a far-reaching generalization of the quadratic reciprocity law.

4.1 The ring of K -integers mod N

Let K be a *number field*, i.e. a finite field extension of the rational numbers \mathbf{Q} , and let \mathcal{O}_K be its ring of integers. We require that during the setup, the primes p, q are chosen to be *unramified* in K , or in other words that they do not divide the discriminant Δ_K . Now, fix a prime number ℓ : we must additionally assume that $\ell \nmid N$. We think of $\ell\Delta_K$ as being a fixed small constant (formally, $\ell\Delta_K$ should be constant with respect to the security parameter), and in particular, $\ell\Delta_K$ should be *much* smaller than the primes p, q , which are of order 2^{1024} . So requiring N and $\ell\Delta_K$ to be coprime to each other is no real constraint.

Next, we make the assumption that K contains a primitive ℓ -th root of unity, that is, that $\mathbf{Q}(\zeta_\ell) \subseteq K$.

Practically, ℓ should be small enough that the discrete logarithm problem in $(\mathbf{Z}/\ell\mathbf{Z})^\times$ is tractable. In fact, taking $\ell = 2$ gives the shortest ciphertexts, as we will have to repeat encryption for ℓ different values. We also need ℓ to be small enough that restricting to primes which are 1 modulo ℓ has negligible effect on the difficulty of the RSA problem.

For concrete security, NIST [1] recommends that RSA public keys N have $\log(N) \geq 2048$. Using the best known factoring algorithms, the difficulty of breaking RSA with such N is approximately

the same as breaking a strong symmetric cipher (such as AES, the industry-standard block cipher) whose key has length 112. Thus, we are looking for primes of size approximately 2^{1024} : by the prime number theorem, the density of primes up to that size is $\frac{1}{1024}$, so taking $\ell \ll 1024$ is sufficient to ensure that the requirement that $p, q \equiv 1 \pmod{\ell}$ does not reduce the security or time required to find N by too large of a margin. As such primes occur with density $\frac{1}{\varphi(\ell)} \geq \frac{1}{\ell}$, the security is reduced at worst from 112 bits to 102 bits.

Now, we discuss how to concretely represent elements of K and its ring of integers \mathcal{O}_K in a form suitable for computation. First, we note that by the primitive element theorem, any number field K is generated as a field over \mathbf{Q} by a single element α . Let $f(x)$ be the monic minimal polynomial of α : by replacing α by $\gamma\alpha$ for some nonzero $\gamma \in \mathbf{Z}$, we may clear the denominators in $f(x)$ to arrange that $f(x) \in \mathbf{Z}[x]$, i.e. that α is an algebraic integer. To wit: we have

$$f(x) = x^d + a_1x^{d-1} + \cdots + a_d, \quad 0 = f(\alpha) = \alpha^d + a_1\alpha^{d-1} + \cdots + a_d$$

Taking γ to be the least common multiple of the denominators appearing in $\{a_1, \dots, a_d\}$, we have:

$$0 = \gamma^d f(\alpha) = (\gamma\alpha)^d + \gamma a_1(\gamma\alpha)^{d-1} + \gamma^2 a_2(\gamma\alpha)^{d-2} + \cdots + \gamma^d a_d$$

As we have $\gamma a_i \in \mathbf{Z}$ for all i , this is a monic polynomial with integer coefficients.

Then we have:

$$K = \mathbf{Q}(\alpha) \simeq \mathbf{Q}[x]/f(x)$$

In other words, we can represent elements of K as \mathbf{Q} -polynomials of degree less than $d = \deg(f(x))$. Now, we consider the subring $\mathbf{Z}[\alpha]$ of K generated over \mathbf{Z} by α . We have

$$\mathbf{Z}[\alpha] \simeq \mathbf{Z}[x]/f(x)$$

So we may represent elements of this ring uniquely as \mathbf{Z} -polynomials of degree less than d . Since α is an algebraic integer, $\mathbf{Z}[\alpha]$ is a subring of the ring of algebraic integers \mathcal{O}_K . As both of these are free \mathbf{Z} -modules in K of rank d , $\mathbf{Z}[\alpha]$ has finite index inside of \mathcal{O}_K . In fact, we have:

$$\mathbf{Z}[\alpha] \subseteq \mathcal{O}_K \subseteq \mathbf{Z}[\alpha, 1/f'(\alpha)] \quad (4.1.1)$$

where $f'(\alpha)$ is the derivative of the polynomial $f(x)$ evaluated at α .

This fact is proven by considering the non-degenerate bilinear pairing on the d -dimensional \mathbf{Q} -vector space K defined by $\langle x, y \rangle = \text{Tr}_{K/\mathbf{Q}}(xy)$, where $\text{Tr}_{K/\mathbf{Q}}(xy)$ is the trace of the linear transformation given by multiplication by xy . As \mathcal{O}_K is a subring of K and the trace of an algebraic

integer is in \mathbf{Z} (indeed, we can compute that the trace is an integer multiple of the highest coefficient of the minimal polynomial), we know that $\langle x, y \rangle \in \mathbf{Z}$ for any $x \in \mathbf{Z}[\alpha], y \in \mathcal{O}_K$. Thus, \mathcal{O}_K is contained inside the dual lattice $\frac{1}{D(\alpha)} \cdot \mathbf{Z}[\alpha]$, where

$$D(\alpha) = \det (\mathrm{Tr}_{K/\mathbf{Q}}(\alpha^i \alpha^j))$$

is the *discriminant* of the trace pairing with respect to the \mathbf{Q} -basis $\{1, \alpha, \alpha^2, \dots, \alpha^{d-1}\}$ of K . Finally, we can compute that this discriminant is equal to $\pm N_{K/\mathbf{Q}} f'(\alpha)$. (Here, $N_{K/\mathbf{Q}}(\cdot)$ is the field norm, mapping an element $\xi \in K$ to the determinant of the \mathbf{Q} -linear map given by multiplication by ξ).

The actual field discriminant Δ_K is obtained as the least common multiple of these $D(\alpha)$'s as α ranges over all elements of \mathcal{O}_K ; in particular, Δ_K divides $\pm N_{K/\mathbf{Q}}(f'(\alpha))$. See [20, Chapter 2] for a careful treatment of these ideas.

We will work in the ring $\Lambda = \mathcal{O}_K[1/(N_{K/\mathbf{Q}}(f'(\alpha)) \cdot \ell)]$. By Equation (4.1.1), we have

$$\Lambda = \mathbf{Z}[\alpha][1/(N_{K/\mathbf{Q}}(f'(\alpha)) \cdot \ell)] \simeq \mathbf{Z}[x, y]/(f(x), f'(\alpha) \cdot \ell \cdot y - 1) \quad (4.1.2)$$

These considerations lead us to amend our choice of p, q : we fix K, α , and ℓ and choose p, q which are congruent to 1 modulo ℓ and are coprime to $N_{K/\mathbf{Q}}(f'(\alpha))$. This is a stronger assumption than the assumption that p, q are unramified in K , but it still only removes a finite number of primes from contention. The benefit of this assumption is the following:

$$A = \Lambda/N \simeq (\mathbf{Z}/N\mathbf{Z})[\alpha, 1/(\Delta_K \cdot \ell)] \simeq (\mathbf{Z}/N\mathbf{Z})[x]/f(x) \quad (4.1.3)$$

In other words, we can represent elements of A as polynomials over $\mathbf{Z}/N\mathbf{Z}$ of degree less than d , and compute multiplication in the ring by polynomial multiplication followed by reduction modulo $f(x)$. For this to be efficiently computable, d should not be too large: as with the choice of ℓ , we think of d as a small constant, e.g. with $d \ll 1024$. In fact, even taking $d < 10$ is likely sufficient to explore various interesting cases (quadratic fields, cyclotomic fields of prime and prime power order, fields with non-abelian Galois group, etc.).

Using the factorization of ideals in \mathcal{O}_K , we can describe the structure of A as follows: Let the factorizations of p, q in \mathcal{O}_K be

$$p\mathcal{O}_K = \mathfrak{p}_1 \cdots \mathfrak{p}_m, \quad q\mathcal{O}_K = \mathfrak{q}_1 \cdots \mathfrak{q}_n.$$

By the assumption that p and q are unramified in K , all of the \mathfrak{p}_i and \mathfrak{q}_j are distinct. We write $\mathbf{F}_{\mathfrak{r}}$ for the field $\mathcal{O}_K/\mathfrak{r}$, where \mathfrak{r} is a nonzero prime ideal of \mathcal{O}_K : this is a finite field extension of \mathbf{F}_r , where $r\mathbf{Z} = \mathfrak{r} \cap \mathbf{Z}$ and $r \in \{p, q\}$. With this notation in hand, we have the following decomposition

of A as a product of fields:

$$A = \prod_{\mathfrak{p}|p} \mathbf{F}_{\mathfrak{p}} \times \prod_{\mathfrak{q}|q} \mathbf{F}_{\mathfrak{q}} \quad (4.1.4)$$

Let's summarize what we have so far:

Definition 4.1.1 (Public parameters: the ring A). As public parameters for our IBE scheme, we fix the following data:

- A number field K , with $d = [K : \mathbf{Q}] \ll 1024$.
- An integral generator of K over \mathbf{Q} with minimal polynomial $f(x) \in \mathbf{Z}[x]$ a monic polynomial of degree d .
- A prime number $\ell \ll 1024$.
- The discriminant $D(\alpha) = N_{K/\mathbf{Q}}(f'(\alpha))$: we further constrain the choice of K and α by requiring $D(\alpha) \ll 1024$.
- A fixed primitive ℓ -th root of unity $\zeta \in K$

Definition 4.1.2 (The Master Key). In the Cocks-Kümmer IBE scheme over the ring $A = \mathcal{O}_K/N\mathcal{O}_K$, the following forms the “master” key pair:

- The *master public key* consists of the data specified in Definition 4.1.1 along with an RSA public key N of bit-length $\lambda = 2048$.
- The *master secret key* consists of the factorization

$$N = \prod_{i=1}^m \mathfrak{p}_i \cdot \prod_{i=1}^n \mathfrak{q}_i$$

of $N\mathcal{O}_K$ into prime ideals in \mathcal{O}_K .

We assume that the factorization of N in \mathcal{O}_K is a given, but note that if we have the factorization $N = pq$ of N in \mathbf{Z} , we may efficiently compute the factorization of N in \mathcal{O}_K . Indeed, for $\rho = p, q$, we have

$$A/\rho = \mathcal{O}_K/\rho \simeq \mathbf{F}_{\rho}[\alpha] \simeq \mathbf{F}_{\rho}[x]/f(x)$$

Thus, finding the prime ideals of \mathcal{O}_K dividing $\rho\mathcal{O}_K$ turns into the problem of factoring the polynomial $f(x)$ in $\mathbf{F}_{\rho}[x]$ into irreducible factors. Unlike the problem of factoring integers, the problem of factoring polynomials over a finite field can be done efficiently even for very large ρ and d . For us, d will be of bounded size, e.g. $d \ll 1024$, but ρ will be large (roughly 2^{1024}). See e.g. [12, §3.4].

4.2 The Kummer Sequence

Now, we will take advantage of the assumption that K contains a primitive ℓ -th root of unity ζ . This allows us to fix an isomorphism $\underline{\mathbf{Z}/\ell\mathbf{Z}} \rightarrow \mu_\ell$ over $\mathcal{O}_K[1/\ell]$ defined by mapping 1 to ζ . Using this isomorphism, we have the ℓ -th power Kummer sequence over $\mathcal{O}_K[1/\ell]$:

$$0 \rightarrow \underline{\mathbf{Z}/\ell\mathbf{Z}} \xrightarrow{\sim} \mu_\ell \rightarrow \mathbf{G}_m \xrightarrow{(\cdot)^\ell} \mathbf{G}_m \rightarrow 0 \quad (4.2.1)$$

This is a short exact sequence of algebraic groups over $\mathcal{O}_K[1/\ell]$. The kernel is étale over $\mathcal{O}_K[1/\ell]$, and thus over Λ . Therefore, upon passing to the base change from Λ to A , the exact sequence (4.2.1) is an isogeny short exact sequence of algebraic groups over A as required by our abstract setup from Chapter 3:

$$0 \rightarrow \underline{\mathbf{Z}/\ell\mathbf{Z}} \rightarrow \mathbf{G}_m \xrightarrow{(\cdot)^\ell} \mathbf{G}_m \rightarrow 0. \quad (4.2.2)$$

Note that the isomorphism $\underline{\mathbf{Z}/\ell\mathbf{Z}} \xrightarrow{\sim} \mu_\ell$ over A is defined using the primitive ℓ -th root of unity ζ in \mathcal{O}_K . More precisely, this isomorphism is defined by mapping 1 to the global section

$$\zeta = ((\zeta_{\mathfrak{p}})_{\mathfrak{p}|p}, (\zeta_{\mathfrak{q}})_{\mathfrak{q}|q}) \in \mu_\ell(A) = \prod_{\mathfrak{p}|p} \mu_\ell(\mathbf{F}_{\mathfrak{p}}) \times \prod_{\mathfrak{q}|q} \mu_\ell(\mathbf{F}_{\mathfrak{q}}) \simeq \prod_{\mathfrak{p}|p} \mathbf{Z}/\ell\mathbf{Z} \times \prod_{\mathfrak{q}|q} \mathbf{Z}/\ell\mathbf{Z},$$

where for a prime ideal \mathfrak{r} of \mathcal{O}_K , $\zeta_{\mathfrak{r}}$ denotes ζ modulo \mathfrak{r} .

4.3 Computing the symbol by factoring

Next, we describe the symbol associated to the isogeny (4.2.2), first using the prime ideal factorization of $N\mathcal{O}_K = (p\mathcal{O}_K)(q\mathcal{O}_K) = \mathfrak{p}_1 \cdots \mathfrak{p}_m \cdot \mathfrak{q}_1 \cdots \mathfrak{q}_m$. Then, we discuss how to use class field theory for K to compute the symbol *without* being able to factor N .

Let \mathfrak{r} be a nonzero prime ideal of A , with $r := \mathfrak{r} \cap \mathbf{Z} \in \{p, q\}$, and choose some $h \in \mathbf{F}_{\mathfrak{r}}^\times$. Recall the computation of the symbol from (3.2.5):

$$\text{sym}_{\mathfrak{r}}(h) = \text{Frob}_{\mathfrak{r}}(\tilde{h}) * \tilde{h}^{-1}.$$

Let $N(\mathfrak{r})$ be the cardinality of $\mathbf{F}_{\mathfrak{r}}$: we have $N(\mathfrak{r}) = r^k$ for some k , with $r \in \{p, q\}$. Here, \tilde{h} is a choice of element $x \in \mathbf{F}^\times$ in some finite field extension \mathbf{F} of $\mathbf{F}_{\mathfrak{r}}$, satisfying $\varphi(x) = x^\ell = h$. As by assumption $p, q \equiv 1 \pmod{\ell}$, so therefore also $N(\mathfrak{r}) \equiv 1 \pmod{\ell}$, we can factor the integer $N(\mathfrak{r}) - 1$ as $N(\mathfrak{r}) - 1 = \ell \cdot (N(\mathfrak{r}) - 1)/\ell$. Now, if \mathbf{F} is a finite field extension of $\mathbf{F}_{\mathfrak{r}}$, the Frobenius map acts on \mathbf{F} over $\mathbf{F}_{\mathfrak{r}}$ by sending x to $x^{N(\mathfrak{r})}$. Since via (4.2.1) the algebraic groups we are considering (G and H from Chapter 3) are both the multiplicative group \mathbf{G}_m , the group multiplication is ordinary multiplication. Thus, we have

$$\text{symb}_{\mathfrak{r}}(h) = (\tilde{h})^{N(\mathfrak{r})-1} = (\tilde{h})^{\ell \cdot \frac{N(\mathfrak{r})-1}{\ell}} = h^{(N(\mathfrak{r})-1)/\ell} \quad (4.3.1)$$

As written above, the local symbol is an element of $\mu_{\ell}(\mathbf{F}_{\mathfrak{r}})$. To get an element of $\mathbf{Z}/\ell\mathbf{Z}$, we use the primitive ℓ -th root of unity $\zeta \in \mathcal{O}_K$: $\text{symb}_{\mathfrak{r}}(h)$ is the unique element $i \in \mathbf{Z}/\ell\mathbf{Z}$ such that

$$\zeta^i \equiv h^{(N(\mathfrak{r})-1)/\ell} \pmod{\mathfrak{r}}. \quad (4.3.2)$$

Finally, multiplying the local symbols for each prime ideal \mathfrak{r} of \mathcal{O}_K dividing N , we can compute the symbol $\text{symb}_{\varphi}(h)$ for $h = (h_{\mathfrak{r}})_{\mathfrak{r}|N\mathcal{O}_K} \in \mathbf{G}_m(A)$ as

$$\text{symb}_{\varphi}(h) = \prod_{\mathfrak{r}|(N\mathcal{O}_K)} \text{symb}_{\mathfrak{r}}(h_{\mathfrak{r}}) \quad (4.3.3)$$

4.4 Kummer theory in terms of Galois groups

We briefly recall the key facts of class field theory, focusing on the formulation of the general power reciprocity law in a form similar to the law of quadratic reciprocity. A good reference that gives a similar explicit algebraic treatment of the higher reciprocity laws is [21, §VIII.5]. There are many good textbook treatments of class field theory. For an extensive treatment of local class field theory and the Hilbert symbol, see [22, Chapter XIII-XIV]. The canonical reference for class field theory is [8], but unfortunately their treatment of the general power residue symbol and reciprocity law is relegated to exercises.

Recall that we are working over a number field K which contains a primitive ℓ -th root of unity ζ , and let $\Gamma = \text{Gal}(K^{\text{alg}}/K)$ be the absolute Galois group of K .

As discussed above, for such fields, the ℓ -th power Kummer sequence - together with the isomorphism $\underline{\mathbf{Z}/\ell\mathbf{Z}} \xrightarrow{\sim} \mu_{\ell}$ determined by ζ - defines an isomorphism

$$H^1(K, \mathbf{Z}/\ell\mathbf{Z}) = \text{Hom}(\Gamma, \mathbf{Z}/\ell\mathbf{Z}) \xleftarrow{\sim} K^{\times} / (K^{\times})^{\ell} \quad (4.4.1)$$

This correspondence works by mapping an element $\alpha \in K^{\times}$ to the character $\chi_{\alpha} : \Gamma \rightarrow \mathbf{Z}/\ell\mathbf{Z}$ defined by $\chi_{\alpha}(\sigma) = \sigma(\sqrt[\ell]{\alpha}) * \sqrt[\ell]{\alpha}^{-1}$, where $\sqrt[\ell]{\alpha}$ is any ℓ -th root of α in the algebraic closure K^{alg} of K . This character is well-defined due to the fact that $\zeta \in K$: choosing another ℓ -th root of α amounts to multiplying the original $\sqrt[\ell]{\alpha}$ by a power of ζ , and since $\zeta \in K$, we have

$$\sigma(\zeta \sqrt[\ell]{\alpha}) * (\zeta \sqrt[\ell]{\alpha})^{-1} = \zeta \sigma(\sqrt[\ell]{\alpha}) * (\zeta \sqrt[\ell]{\alpha})^{-1} = \sigma(\sqrt[\ell]{\alpha}) * \sqrt[\ell]{\alpha}^{-1}.$$

As written, χ_{α} is valued in $\mu_{\ell}(K)$, but the choice of ζ allows us to consider χ_{α} as valued in $\mathbf{Z}/\ell\mathbf{Z}$: $\chi_{\alpha}(\sigma)$ is the value of $i \in \mathbf{Z}/\ell\mathbf{Z}$ such that $\sigma(\sqrt[\ell]{\alpha}) = \zeta^i \sqrt[\ell]{\alpha}$. Thus, we see that the character χ_{α} defines an isomorphism between the Galois group of the extension $K(\sqrt[\ell]{\alpha})/K$ and $\mathbf{Z}/\ell\mathbf{Z}$. The fact that the

map in (4.4.1) is an isomorphism implies that *every* extension of K with Galois group isomorphic to $\mathbf{Z}/\ell\mathbf{Z}$ arises this way, i.e. is the extension obtained by adjoining an ℓ -th root of an element of K^\times , well-defined up to multiplication by ℓ -th powers.

Class field theory provides a description of all abelian extensions of K in terms of the arithmetic of \mathcal{O}_K . The general formulation is rather abstract, but the theory of the Kummer sequence allows us to be more explicit in the context of cyclic degree ℓ extensions of a number field containing a primitive ℓ -th root of unity, as we turn to now.

4.5 The Hilbert symbol

Just like quadratic reciprocity, the general power reciprocity law is formulated in terms of “symbols”, i.e. functions

$$K^\times / (K^\times)^\ell \times K^\times / (K^\times)^\ell \rightarrow \mathbf{Z}/\ell\mathbf{Z}$$

which are multiplicative in both variables. These symbols are first defined prime-by-prime (“locally”), then multiplied together to obtain a “global” symbol, just as Legendre symbols are multiplied together to obtain Jacobi symbols.

These local symbols are called *Hilbert symbols*. The canonical reference for Hilbert symbols is [22, Chapter XIV]. There is also a very explicit algebraic treatment in [21, §VIII.5].

Definition 4.5.1 (Hilbert symbol). Let \mathfrak{r} be a nonzero prime ideal of K . For each such \mathfrak{r} , we have a local *degree ℓ Hilbert symbol*

$$(\cdot, \cdot)_{\mathfrak{r}}: K^\times / (K^\times)^\ell \times K^\times / (K^\times)^\ell \rightarrow \mathbf{Z}/\ell\mathbf{Z}$$

Given two elements $a, b \in K^\times$, we may apply the Kummer isomorphism to obtain characters $\chi_a, \chi_b: \Gamma \rightarrow \mathbf{Z}/\ell\mathbf{Z}$. Let $\Gamma_{\mathfrak{r}}$ be a *decomposition group* for \mathfrak{r} (a subgroup of Γ that fixes a choice of prime lying above \mathfrak{r}). We may restrict the characters χ_a, χ_b to $\Gamma_{\mathfrak{r}}$, and consider these as cohomology classes in $H^1(\Gamma_{\mathfrak{r}}, \mathbf{Z}/\ell\mathbf{Z})$. Then we have:

$$(a, b)_{\mathfrak{r}} = \text{inv}_{\mathfrak{r}}(\chi_a \cup \chi_b), \tag{4.5.1}$$

with $\text{inv}_{\mathfrak{r}}: H^2(\Gamma_{\mathfrak{r}}, \mathbf{Z}/\ell\mathbf{Z}) \xrightarrow{\sim} \mathbf{Z}/\ell\mathbf{Z}$ the local *invariant map*. Here, $\chi_a \cup \chi_b$ is considered as an element of $H^2(\Gamma_{\mathfrak{r}}, \mathbf{Z}/\ell\mathbf{Z})$. The choice of primitive ℓ -th root of unity ζ allows us to identify $H^2(\Gamma_{\mathfrak{r}}, \mathbf{Z}/\ell\mathbf{Z})$ with $H^2(\Gamma_{\mathfrak{r}}, \mu_\ell)$. Local class field theory provides an interpretation of this cohomology group in terms of algebraic data associated to the local field $K_{\mathfrak{r}}$ (the completion of K in the \mathfrak{r} -adic topology), and furnishes us with the *invariant map*

$$\text{inv}_{\mathfrak{r}}: H^2(\Gamma_{\mathfrak{r}}, \mu_\ell) \xrightarrow{\sim} \mathbf{Z}/\ell\mathbf{Z}$$

Milne's class field theory notes [21, §III.1, §III.4, §VIII.5] provides a thorough treatment of this construction in terms of explicit algebra, and Serre [22, Chapter XIV] gives a comprehensive treatment of the cohomology interpretation of the theory.

We have the following key computations:

Theorem 4.5.2. *Suppose that $a, b \in \mathcal{O}_K - \{0\}$ are both coprime to $\ell\mathcal{O}_K$. Then we have:*

- $(a, b)_{\mathfrak{r}} = (b, a)_{\mathfrak{r}}^{-1}$.
- $(a, b)_{\mathfrak{r}}$ is a homomorphism in both arguments; that is:

$$(a_1, b)_{\mathfrak{r}}(a_2, b)_{\mathfrak{r}} = (a_1 a_2, b)_{\mathfrak{r}}; \quad (a, b_1)_{\mathfrak{r}}(a, b_2)_{\mathfrak{r}} = (a, b_1 b_2)_{\mathfrak{r}}$$

- If $a \notin \mathfrak{r}$ and $\ell \notin \mathfrak{r}$, we have

$$(a, b)_{\mathfrak{r}} = \chi_a(\text{Frob}_{\mathfrak{r}})^{\text{ord}_{\mathfrak{r}}(b)} = \left(a^{(N(\mathfrak{r})-1)/\ell}\right)^{\text{ord}_{\mathfrak{r}}(b)} = \text{symb}_{\mathfrak{r}}(a)^{\text{ord}_{\mathfrak{r}}(b)}, \quad (4.5.2)$$

with $N(\mathfrak{r})$ the cardinality of $\mathbf{F}_{\mathfrak{r}}$.

- In particular, if $\text{ord}_{\mathfrak{r}}(b) = 1$, $(a, b)_{\mathfrak{r}} = 1$ exactly when a is an ℓ -th power modulo \mathfrak{r} .
- For any $a, b \in K^{\times}$, $(a, b)_{\mathfrak{r}} = 2$ for all \mathfrak{r} which are coprime to a, b , and ℓ . In particular, $(a, b)_{\mathfrak{r}} = 1$ for all but finitely many \mathfrak{r} .

Proof. See [21, Chapter III, Theorem 4.4] or [22, Chapter XIV, §2, Proposition 4]. To prove the final equality in (4.5.2), observe that the term $a^{(\mathfrak{r}-1)/\ell}$ appearing in (4.5.2) agrees precisely with the formula for $\text{symb}_{\mathfrak{r}}(a)$ given in (4.3.1). □

To complete the formulation of the general reciprocity laws, we need one more piece of notation. In addition to the set of prime ideals of \mathcal{O}_K , we must consider the embeddings of K into the complex numbers \mathbf{C} up to complex conjugation. More precisely, we consider either embeddings of K into the real numbers \mathbf{R} or pairs of distinct complex-conjugate embeddings of K into \mathbf{C} whose image is not contained in \mathbf{R} . These can be thought of as “points at infinity” for the affine “curve” $\text{Spec}(\mathcal{O}_K)$. The definition of the Hilbert symbol can be understood in this context as well. If K admits an embedding into \mathbf{R} , then complex conjugation restricts to an involution of K^{alg} fixing K : we can restrict the Galois characters χ_a, χ_b to this subgroup and consider the invariant of $\chi_a \cup \chi_b \in H^2(\text{Gal}(\mathbf{C}/\mathbf{R}), \mathbf{Z}/\ell\mathbf{Z})$. This result is quite simple, as the absolute Galois group of the real numbers is just a cyclic group of order 2, generated by complex conjugation. We end up with the following:

Definition 4.5.3. Let $\sigma: K \hookrightarrow \mathbf{R}$ be an embedding of K into the real numbers. Then we have the *archimedean Hilbert symbol*

$$(a, b)_\sigma = \begin{cases} -1 & \sigma(a) < 0 \text{ and } \sigma(b) < 0 \\ 1 & \text{otherwise} \end{cases}$$

Note that the archimedean symbols are only relevant in the case $\ell = 2$: otherwise, by the assumption that K contains a primitive ℓ -th root of unity, it is not possible to embed K into \mathbf{R} .

4.6 The Artin reciprocity law

The local theory of Hilbert symbols in place, we can state the main theorem of class field theory: the Artin reciprocity law.

Theorem 4.6.1. *Let K be a number field containing a primitive ℓ -th root of unity ζ . Then for $a, b \in K^\times$, we have:*

$$\prod_{\mathfrak{r} \in |\text{MaxSpec}(\mathcal{O}_K)|} (a, b)_{\mathfrak{r}} \times \prod_{\sigma: K \hookrightarrow \mathbf{R}} (a, b)_\sigma = 1 \quad (4.6.1)$$

Here, $\text{MaxSpec}(\mathcal{O}_K)$ refers to the set of *maximal* ideals of \mathcal{O}_K . These are the same as the *non-zero* prime ideals.

All known proofs of this result are quite involved: see e.g. [8, VII.10] or [21, V.3] for statements of the general Artin reciprocity law in a somewhat different form. For the statement above, see [21, Theorem 5.11, Chapter V].

Definition 4.6.2. Suppose $a, b \in K^\times$ are coprime to each other and suppose that b is coprime to $\ell\mathcal{O}_K$. Then, we define the *power reciprocity symbol* as:

$$\left(\frac{a}{b}\right)_{K, \ell} = \prod_{\text{ord}_{\mathfrak{r}}(b) \neq 0} (a, b)_{\mathfrak{r}}^{\text{ord}_{\mathfrak{r}}(b)}$$

Note that if $\ell = 2$ and a, b are odd positive integers, this is exactly the Jacobi symbol. Moreover, setting $b = N$ and using the computation of the symbol symb_φ from (4.3.3), we observe:

Theorem 4.6.3. *The symbol symb_φ of our IBE scheme from Sections 4.1 and 4.2 is exactly the power residue symbol. Namely:*

$$\text{symb}_\varphi(a) = \left(\frac{a}{N}\right)_{K, \ell} \quad (4.6.2)$$

Proof. Recall from above that we derived (4.3.3) and (4.3.1), describing the symbol locally by factoring N . Now, applying these computations along with (4.5.2), we see that:

$$\text{symb}_\varphi(a) = \prod_{\mathfrak{r} | (N\mathcal{O}_K)} \text{symb}_{\mathfrak{r}}(a_{\mathfrak{r}}) = \prod_{\mathfrak{r} | (N\mathcal{O}_K)} a_{\mathfrak{r}}^{(\mathfrak{r}-1)/\ell} = \prod_{\mathfrak{r} | (N\mathcal{O}_K)} (a, N)_{\mathfrak{r}} = \left(\frac{a}{N}\right)_{K, \ell}$$

Note that by the assumption that p, q are unramified in K , for any prime \mathfrak{r} dividing $N\mathcal{O}_K$, we have $\text{ord}_{\mathfrak{r}}(N) = 1$. \square

Using the above computation of Hilbert symbols, we can restate the Artin reciprocity law as:

Theorem 4.6.4. *Let $a, b \in K^\times$ be coprime to $\ell\mathcal{O}_K$ and to each other. Then we have:*

$$\left(\frac{a}{b}\right)_{K,\ell} \cdot \left(\frac{b}{a}\right)_{K,\ell}^{-1} \cdot \prod_{\mathfrak{r}|\ell\mathcal{O}_K} (a, b)_{\mathfrak{r}} \cdot \prod_{\sigma: K \hookrightarrow \mathbf{R}} (a, b)_{\sigma} = 1$$

Proof. Let's see how to derive this from (4.6.1). First, note that the term involving the embeddings of K into \mathbf{R} is the same in both equations, so we may ignore it. Now, we can partition the prime ideals of \mathcal{O}_K into four sets $\{S_0, S(a), S(b), S(\ell)\}$, where for $x \in K^\times$, $S(x)$ denotes the set of nonzero prime ideals \mathfrak{r} such that $\text{ord}_{\mathfrak{r}}(x) \neq 0$, and S_0 denotes the set of primes such that $\text{ord}_{\mathfrak{r}}(a) = \text{ord}_{\mathfrak{r}}(b) = \text{ord}_{\mathfrak{r}}(\ell) = 0$. Note that our assumptions on a, b , and ℓ imply that these sets are *disjoint*. We describe the Hilbert symbol for each of these sets:

- If $\mathfrak{r} \in S_0$, then $(a, b)_{\mathfrak{r}} = 1$ by Theorem 4.5.2.
- If $\mathfrak{r} \in S(b)$, then $\text{ord}_{\mathfrak{r}}(a) = \text{ord}_{\mathfrak{r}}(\ell) = 0$, so we may apply the formula for the unramified symbol from Theorem 4.5.2:

$$\prod_{\mathfrak{r} \in S(b)} (a, b)_{\mathfrak{r}} = \prod_{\mathfrak{r}|\text{ord}_{\mathfrak{r}}(b) \neq 0} \chi_a(\text{Frob}_{\mathfrak{r}})^{\text{ord}_{\mathfrak{r}}(b)} = \left(\frac{a}{b}\right)_{K,\ell} \quad (4.6.3)$$

- If $\mathfrak{r} \in S(a)$, then $\text{ord}_{\mathfrak{r}}(b) = 0$ and $\ell \notin \mathfrak{r}$. From Theorem 4.5.2, we know that $(a, b)_{\mathfrak{r}} = (b, a)_{\mathfrak{r}}^{-1}$. Thus, we have

$$\prod_{\mathfrak{r} \in S(a)} (a, b)_{\mathfrak{r}} = \prod_{\mathfrak{r} \in S(a)} (b, a)_{\mathfrak{r}}^{-1} = \prod_{\mathfrak{r} \in S(b)} \chi_b(\text{Frob}_{\mathfrak{r}})^{-\text{ord}_{\mathfrak{r}}(a)} = \left(\frac{b}{a}\right)_{K,\ell}^{-1} \quad (4.6.4)$$

- If $\mathfrak{r} \in S(\ell)$, the Hilbert symbol is “wild” and can be more complicated to compute. However, the computation is “small” in the sense that $(a, b)_{\mathfrak{r}}$ for $\mathfrak{r}|\ell$ only depends on the values of a, b modulo $\ell^f\mathcal{O}_K$ for some integer f depending only on ℓ and K . As ℓ is held constant and much smaller than N , we may regard this computation as a “constant time” operation - assuming we have a method to compute it at all! We'll say a little bit more about this in the next section.

Now, as the sets $\{S_0, S(a), S(b), S(\ell)\}$ form a disjoint partition of $|\text{MaxSpec}\mathcal{O}_K|$, the Artin reciprocity law (4.6.1) can be rewritten as:

$$\prod_{\mathfrak{r} \in S_0} (a, b)_{\mathfrak{r}} \times \prod_{\mathfrak{r} \in S(b)} (a, b)_{\mathfrak{r}} \times \prod_{\mathfrak{r} \in S(a)} (a, b)_{\mathfrak{r}} \times \prod_{\mathfrak{r} \in S(\ell)} (a, b)_{\mathfrak{r}} \times \prod_{\sigma: K \hookrightarrow \mathbf{R}} (a, b)_{\sigma} = 1$$

By the above computation of the Hilbert symbol in these four cases, we can rewrite this as:

$$1 \times \left(\frac{a}{b}\right)_{K,\ell} \times \left(\frac{b}{a}\right)_{K,\ell}^{-1} \times \prod_{\mathfrak{r} | (\ell \mathcal{O}_K)} (a, b)_{\mathfrak{r}} \times \prod_{\sigma: K \hookrightarrow \mathbf{R}} (a, b)_{\sigma} = 1$$

This is exactly the statement of Theorem 4.6.4. \square

We can rearrange this statement to the following form, which suggests the method for computing $\left(\frac{a}{b}\right)_{K,\ell}$:

$$\left(\frac{a}{b}\right)_{K,\ell} = \left(\frac{b}{a}\right)_{K,\ell}^{-1} \cdot \prod_{\mathfrak{r} | (\ell \mathcal{O}_K)} (b, a)_{\mathfrak{r}} \cdot \prod_{\sigma: K \hookrightarrow \mathbf{R}} (b, a)_{\sigma} \quad (4.6.5)$$

4.7 Computing the power residue symbol

Finally, we discuss the problem of efficiently computing the power residue symbol without knowledge of the factorization of N , following [14] and [13]. See also [19] for the genesis of some of the ideas involved. The full method for doing so is rather involved, and therefore we will only give an outline of some of the key ideas.

At the heart of the method is, of course, the power reciprocity law. Looking at (4.6.5), we see that in order to perform the sort of “repeated swapping” Euclidean reduction that we successfully used in the quadratic case over $K = \mathbf{Q}$ in Theorem 1.2.1, we need to be able to efficiently compute the “wild” Hilbert symbols $(a, b)_{\mathfrak{r}}$ for $\mathfrak{r} | \ell \mathcal{O}_K$ (as well as the archimedean symbols in the case $\ell = 2$). Doing so requires delicate computations in the arithmetic of ℓ -adic completions of K : in particular, how arithmetic operations interact with the filtration of units u by $\text{ord}_{\mathfrak{r}}(u - 1)$. The details of this computation are covered in [7, Chapter 5] and [13, §3.5].

Taking the ability to efficiently compute Hilbert symbols as a black box, the algorithm described in [13, Chapter 4] proceeds similarly to the classical algorithm for computing the Jacobi symbol as in Theorem 1.2.1. As a reminder, throughout this work, an algorithm is “efficient” if it runs in time polynomial in $\log N$, (or equivalently, polynomial in the security parameter λ). However, the existence of the Euclidean division algorithm and the definitive notion of size provided by the ordinary absolute value on \mathbf{Z} are very special properties of the rational integers, and very rarely extend to the rings of integers of other number fields. This means that we can’t really make sense of the idea of “reducing a modulo b ”, to obtain a canonical “smallest” representative of the modular equivalence class. Because of this, a somewhat different technique is required.

The algorithm of [13] computes $\left(\frac{\alpha}{\beta}\right)_{K,\ell}$ by replacing α by $\alpha \cdot \gamma^{\ell} + \beta \cdot \delta$ for various nonzero $\gamma, \delta \in \mathcal{O}_K$ until we obtain an element α' such that $\alpha' \mathcal{O}_K$ is the product of a single large prime ideal with a number of small (i.e. with norm less than some fixed bound) prime ideals. By construction, we will have $\left(\frac{\alpha}{\beta}\right)_{K,\ell} = \left(\frac{\alpha'}{\beta}\right)_{K,\ell}$. Furthermore, it is possible to *efficiently compute the prime factorization* of $\alpha' \mathcal{O}_K$! Then, using the power reciprocity law along with the algorithm of [7, Chapter 5] to compute

Hilbert symbols, we can reduce the problem of calculating $\left(\frac{\alpha}{\beta}\right)_{K,\ell} = \left(\frac{\alpha'}{\beta}\right)_{K,\ell}$ to the problem of computing $\left(\frac{\beta}{\alpha'}\right)_{K,\ell}$, which can now be computed directly from Definition 4.6.2 using the prime factorization of $\alpha' \mathcal{O}_K$.

4.8 Coda: Recovering the BLS construction

How does the construction of [5] fit into our picture? In their construction, they work with the ℓ -th power map on $(\mathbf{Z}/N\mathbf{Z})^\times$ and use the ℓ -th power reciprocity symbol. However, this construction does not quite fit into our schema as stated: since \mathbf{Q} does not contain primitive ℓ -th roots of unity for $\ell > 2$, in order to use the ℓ -th power reciprocity symbol we work in the field $K = \mathbf{Q}(\zeta_\ell)$ for ζ_ℓ some primitive ℓ -th root of unity.

However, for this field, $\mathcal{O}_K/N\mathcal{O}_K$ is of course larger than $\mathbf{Z}/N\mathbf{Z}$. To reconcile our notion with [5], we note that [5] assumes that p, q are 1 modulo ℓ , and the construction in [5] relies on the choice of a primitive ℓ -th root of unity ζ in $\mathbf{Z}/N\mathbf{Z}$. We have $\mathcal{O}_{\mathbf{Q}(\zeta_\ell)} \simeq \mathbf{Z}[X]/\Phi_\ell(X)$ with $\Phi_\ell(X) = (X^\ell - 1)/(X - 1) = 1 + X + X^2 + \cdots + X^{\ell-1}$ the cyclotomic polynomial. (This formula written here for the cyclotomic polynomial is only correct for the case that ℓ is prime, as we assume everywhere in this thesis; however, the BLS paper does not make this assumption.) From this, we deduce that a $\mathbf{Z}/N\mathbf{Z}$ -algebra section of $\mathcal{O}_{\mathbf{Q}(\zeta_\ell)}/N\mathcal{O}_{\mathbf{Q}(\zeta_\ell)}$ is equivalent to a choice of $\zeta \in \mathbf{Z}/N\mathbf{Z}$ such that $\Phi_\ell(\zeta) = 0$; i.e. to the choice of a primitive ℓ -th root of unity. Such a ζ exists if and only if p, q are congruent to 1 modulo ℓ .

In other words, the assumptions on p, q and choice of ζ in [5] amount to realizing $\mathbf{Z}/N\mathbf{Z}$ as $\mathcal{O}_K/\mathfrak{b}$ for some square-free ideal \mathfrak{b} . As such, the power reciprocity symbol of interest ends up being

$$\text{symb}_\varphi(a) = \left(\frac{a}{\mathfrak{b}}\right)_{\mathbf{Q}(\zeta_\ell),\ell} = \prod_{\mathfrak{r}|\mathfrak{b}} \left(\frac{a}{\mathfrak{r}}\right)_{\mathbf{Q}(\zeta_\ell),\ell}^{\text{ord}_{\mathfrak{r}}(\mathfrak{b})}$$

We could have done our whole construction in this chapter more generally, replacing $\mathcal{O}_K/N\mathcal{O}_K$ with $A = \mathcal{O}_K/\mathfrak{b}$ everywhere with little change (all of the results of Chapter 3 apply equally well in this case). While the discussion of the power reciprocity symbols becomes more awkward, as \mathfrak{b} may fail to be a principal ideal, the difficulty is only expositional. The results of [14] and [13] allow for efficient computation of $\left(\frac{a}{\mathfrak{b}}\right)_{\mathbf{Q}(\zeta_\ell),\ell}$ for any nonzero ideal \mathfrak{b} which is coprime to ℓ . The computation still uses Theorem 4.6.4 (the Artin reciprocity law) in a crucial way: as a first step of the computation of $\left(\frac{a}{\mathfrak{b}}\right)_{\mathbf{Q}(\zeta_\ell),\ell}$, we search for some $b \in K^\times$ such that $\mathfrak{b} * b^{-1}$ is easy to factor. Then, we have

$$\left(\frac{a}{\mathfrak{b}}\right)_{\mathbf{Q}(\zeta_\ell),\ell} = \left(\frac{a}{\mathfrak{b} * b^{-1}}\right)_{\mathbf{Q}(\zeta_\ell),\ell} * \left(\frac{a}{b}\right)_{\mathbf{Q}(\zeta_\ell),\ell}$$

Since $\mathfrak{b} * b^{-1}$ can be efficiently factored, we compute the first factor using this factorization as in Section 4.3. $\left(\frac{a}{b}\right)_{\mathbf{Q}(\zeta_\ell),\ell}$, thereby reducing to the case where \mathfrak{b} is a principal ideal.

Bibliography

- [1] Elaine B. Barker and Quynh H. Dang. Recommendation for key management part 3: Application-specific key management guidance. Technical report, January 2015.
- [2] Dan Boneh and Matt Franklin. Identity-Based Encryption from the Weil Pairing. In Joe Kilian, editor, *Advances in Cryptology — CRYPTO 2001*, Lecture Notes in Computer Science, pages 213–229. Springer, 2001.
- [3] Dan Boneh, Craig Gentry, and Michael Hamburg. Space-Efficient Identity Based Encryption Without Pairings, 2007.
- [4] Dan Boneh and Rio LaVigne. Identity-Based Encryption with e^{th} Residuosity and its Incompressibility. page 12, 2013.
- [5] Dan Boneh, Rio LaVigne, and Manuel Sabin. Identity-based encryption with e^{th} residuosity and its incompressibility. In *Autumn 2013 TRUST Conference. Washington DC (Oct 9-10, 2013), Poster Presentation*, 2013.
- [6] Dan Boneh and Victor Shoup. A Graduate Course in Applied Cryptography.
- [7] Johannes Bouw. *On the computation of norm residue symbols*. PhD thesis, Leiden University, 2021. Available at <https://scholarlypublications.universiteitleiden.nl/handle/1887/3176464>.
- [8] J.W.S. Cassels and A. Fröhlich. *Algebraic Number Theory: Proceedings of an Instructional Conference Organized by the London Mathematical Society (a NATO Advanced Study Institute) with the Support of the International Mathematical Union*. London Mathematical Society, 2010.
- [9] Sanjit Chatterjee and Palash Sarkar. *Identity-Based Encryption*. Springer US, 2011.
- [10] Michael Clear and Ciaran McGoldrick. Additively Homomorphic IBE from Higher Residuosity. In Dongdai Lin and Kazue Sako, editors, *Public-Key Cryptography – PKC 2019*, pages 496–515. Springer International Publishing, 2019.

- [11] Clifford Cocks. An Identity Based Encryption Scheme Based on Quadratic Residues. In Bahram Honary, editor, *Cryptography and Coding*, Lecture Notes in Computer Science, pages 360–363. Springer, 2001.
- [12] Henri Cohen. *A Course in Computational Algebraic Number Theory*, volume 138 of *Graduate Texts in Mathematics*. Springer, 1993.
- [13] Koen de Boer. *Computing the power residue symbol*. PhD thesis, Master’s thesis. Nijmegen, Radboud University. www.koendeboer.com, 2016.
- [14] Koen de Boer and Carlo Pagano. Calculating the power residue symbol and ibeta: Applications of computing the group structure of the principal units of a p-adic number field completion. In *Proceedings of the 2017 ACM on International Symposium on Symbolic and Algebraic Computation*, ISSAC ’17, page 117–124, New York, NY, USA, 2017. Association for Computing Machinery.
- [15] Kenneth Ireland and Michael Rosen. *A Classical Introduction to Modern Number Theory*, volume 84 of *Graduate Texts in Mathematics*. Springer New York, 1990.
- [16] Gábor Ivanyos, Marek Karpinski, Lajos Rónyai, and Nitin Saxena. Trading Grh for Algebra: Algorithms for Factoring Polynomials and Related Structures. 81(277):493–531.
- [17] Marc Joye. Identity-Based Cryptosystems and Quadratic Residuosity. In Chen-Mou Cheng, Kai-Min Chung, Giuseppe Persiano, and Bo-Yin Yang, editors, *Public-Key Cryptography – PKC 2016*, Lecture Notes in Computer Science, pages 225–254. Springer, 2016.
- [18] Donald Ervin Knuth. *The Art of Computer Programming*. Addison-Wesley Series in Computer Science and Information Processing. Addison-Wesley Pub. Co, 1973.
- [19] HW Lenstra. Computing jacobi symbols in algebraic number fields. *Nieuw Arch. Wisk.*, 13:421–426, 1995.
- [20] James S. Milne. Algebraic number theory (v3.08), 2020. Available at www.jmilne.org/math/.
- [21] J.S. Milne. Class field theory (v4.03), 2020. Available at www.jmilne.org/math/.
- [22] Jean-Pierre Serre. *Local fields*, volume 67. Springer Science & Business Media, 2013.
- [23] Adi Shamir. Identity-based cryptosystems and signature schemes. In *Workshop on the Theory and Application of Cryptographic Techniques*, pages 47–53. Springer, 1984.
- [24] UkoeHB. Mechanics-of-MobileCoin (Preview 10/11 chapters), 2022-01.