Primeless Cryptography

Sonia Mihaela Bogos

School of Computer and Communication Sciences

Summer Internship

September 2011

Responsible
Prof. Serge Vaudenay
EPFL / LASEC

Supervisor
Ioana Boureanu
EPFL / LASEC

Contents

1 Introduction and Motivation

The goal of this project is to study and construct cryptographic primitives in a primeless manner, i.e. without generating prime numbers.

The motivation for this approach is to reduce the asymptotic complexity of a primitive that is, in many cases, dominated by the generation of prime numbers. For example, the RSA cryptosystem has a complexity of $O(l^4)$ for the setup phase, for generating two different prime numbers p and q. The encryption and decryption processes take only $O(l^3)$.

Although we are working with random numbers n, we study the problem of how big should be the size of n as we want to maintain the factorization problem hard.

The report is organized as follows: **Background** section introduces the definition of a *character* and presents the computational problems related to it. Based on generation of random numbers and on the properties of a character we propose in this report a scheme of key agreement and an encryption scheme in section **New Proposal**. Another probabilistic scheme was introduced in [3], but we prove, in section **Probabilistic Encryption Scheme**, that it is insecure. **Conclusions** ends the report with some final remarks and future work.

2 Background

In this section, we firstly introduce the definition of a *character* and present the properties for characters of order 2 and 4. Secondly, we summarise the computational problems related to our work. For details, see [2] and [7].

Definition 1. *Let G be an Abelian group. A character χ is a group homomorphism from $(G, +)$ to $(\mathbb{C}\backslash\{0\})$, i.e. a function $\chi : G \to \mathbb{C}\backslash\{0\}$ such that*

$$\chi(a + b) = \chi(a) \cdot \chi(b),\ \forall a, b \in G.$$

From the definition we can deduce that:

1. $\chi(0) = 1$.
2. $\chi(a)$ is the $\lambda(G)$th root of the unity, where $\lambda(G)$ is the group exponent of G.
3. $\forall a \in G,\ \chi(-a) = \chi(a)^{-1}$.

Using these properties, it is possible to define a group structure over the set of all characters on G. The product $\chi_1\chi_2$ of two characters χ_1 and χ_2 is defined as:

$$\chi_1\chi_2(a) = \chi_1(a) \cdot \chi_2(a),\ \forall a \in G.$$

The inverse χ^{-1} is defined by:

$$\chi^{-1} = \chi(a)^{-1},\ \forall a \in G.$$

Each character will have an order in this group. Here we focus on characters of order 2 and 4.

2.1 Characters of order 2

Let ε be the trivial character where $\varepsilon(a) = 1$, $\forall a \in \mathbb{Z}_p^*$. Let $p \in \mathbb{Z}$ be an odd prime. The only two characters for which $\chi^2 = \varepsilon$ are: Legendre symbol, $(\frac{a}{p})$, and ε.

We have the following properties for the Legendre symbol:

Proposition 1. *Let p, q be odd primes and $a, b \in \mathbb{Z}$. Then:*

1. $a^{(p-1)/2} = (\frac{a}{p}) \pmod p$.
2. $(\frac{ab}{p}) = (\frac{a}{p}) \cdot (\frac{b}{p})$.
3. *If $a \equiv b \pmod p$, then $(\frac{a}{p}) = (\frac{b}{p})$.*
4. *(Law of Quadratic Reciprocity)* $(\frac{p}{q})(\frac{q}{p}) = (-1)^{((p-1)/2)((q-1)/2)}$.

The Jacobi symbol is an extension of the Legendre symbol that is defined for any odd integer n:

$$(\tfrac{a}{n}) = (\tfrac{a}{p_1})^{e_1}(\tfrac{a}{p_2})^{e_2}\cdots(\tfrac{a}{p_i})^{e_i},$$

where $gcd(a, n) = 1$ and $n = p_1^{e_1}p_2^{e_2}\cdots p_i^{e_i}$.

For the particular case when $n = p \cdot q$ with p and q two different odd primes, we have a list of four characters, solutions of $\chi^2 = \varepsilon$, in $\mathbb{Z}_n^* : \varepsilon$, $(\frac{\cdot}{n})$, $(\frac{\cdot}{p})$ and $(\frac{\cdot}{q})$.

2.2 Characters of order 4

Characters of order 4 are defined on the ring of Gaussian integers, the set $\mathbb{Z}[i] = \{a + bi \mid a, b \in \mathbb{Z}, i^2 = -1\}$ with the classical operations defined on complex numbers.

For $\alpha \in \mathbb{Z}[i]$, with $\alpha = a + bi$, we define the norm of α as $N(\alpha) = \alpha \cdot \bar{\alpha} = a^2 + b^2$, where $\bar{\alpha} = a - bi$ is the conjugate of α. The units of $\mathbb{Z}[i]$ are 1, -1, i and $-i$.

Proposition 2. *The following statements describe all primes of $\mathbb{Z}[i]$ (up to a unit).*

1. *An element $\alpha \in \mathbb{Z}[i]$ with $N(\alpha)$ equal to a prime p of $\mathbb{Z}$ such that $p \equiv 1 \pmod 4$. Conversely, let $p \in \mathbb{Z}$ be a prime such that $p \equiv 1 \pmod 4$. There exists a prime $\alpha \in \mathbb{Z}[i]$ satisfying*

$$N(\alpha) = \alpha\bar{\alpha} = p.$$

2. *An element $\alpha = p \in \mathbb{Z}$ such that p is a prime of $\mathbb{Z}$ and $p \equiv 3 \pmod 4$.*

3. *$1 + i$ and its conjugate, $1 - i$, are prime and $N(1 + i) = N(1 - i) = 2$.*

Given a prime $p \in \mathbb{Z}$, where $p \equiv 1 \pmod 4$, we can apply the Cornacchia and Tonneli algorithms to find the prime α of $\mathbb{Z}[i]$ with $N(\alpha) = p$. For more details, see [1].

In order to define the extension of the Jacobi symbol for characters of order 4 we need the following result:

Proposition 3. *Let α be a prime of $\mathbb{Z}[i]$. Then, $\mathbb{Z}[i]/\alpha\mathbb{Z}[i]$ is a finite field with $N(\alpha)$ elements.*

Quartic Residue Symbol

Definition 2. *Let $\alpha, \beta \in \mathbb{Z}[i]$ be such that $(1 + i) \nmid \beta$. Values β and α are relatively prime. The quartic residue symbol is defined as $\chi_\beta : \mathbb{Z}[i] \to \{\pm 1, \pm i\}$*

- *for β prime in $\mathbb{Z}[i]$, $\chi_\beta(\alpha)$ is the only element of $\{\pm 1, \pm i\}$ such that $\chi_\beta(\alpha) = (\alpha^{\frac{N(\beta)-1}{4}}) \bmod \beta$.*
- *for $\beta = \prod_i \beta_i$, where β_i is prime in $\mathbb{Z}[i]$, $\chi_\beta(\alpha) = \prod_i \chi_{\beta_i}(\alpha)$.*

From the definition it can be seen that for β prime in $\mathbb{Z}[i]$, $\chi_\beta(\alpha) = 1$ iff $\alpha \equiv z^4 \pmod{\beta}$ has a solution in $\mathbb{Z}[i]^*$.

Notice that for $\beta \in \mathbb{Z}$, for any $\alpha \in \mathbb{Z}$, $\chi_\beta(\alpha) = (\frac{\alpha}{\beta})$.

We introduce the notion of primary element as it is useful in presenting properties of the quartic residue symbol:

Definition 3. *A nonunit $\alpha \in \mathbb{Z}[i]$ is primary if $\alpha \equiv 1 \pmod{(1+i)^3}$. Writing α as $a + bi$, the above congruence is equivalent to*

$$a \equiv 1 \pmod 4,\ b \equiv 0 \pmod 4 \text{ or}$$
$$a \equiv 3 \pmod 4,\ b \equiv 2 \pmod 4.$$

Proposition 4. *For any $\alpha \in \mathbb{Z}[i]$ there is a unique representation of the form $i^j \cdot (1+i)^k \cdot \alpha'$, with $\alpha' \in \mathbb{Z}[i]$ primary and $k \in \mathbb{Z}$, $j \in \{0,1,2,3\}$.*

The quartic residue symbol satisfies the following properties:

Proposition 5. *Let $\alpha, \beta \in \mathbb{Z}[i]$ be such that $(1+i) \nmid \beta$, with β, α and α' relatively prime.*

1. *(Multiplicativity) $\chi_\beta(\alpha\alpha') = \chi_\beta(\alpha) \cdot \chi_\beta(\alpha')$.*
2. *(Modularity) If $\alpha \equiv \alpha' \pmod{\beta}$, $\chi_\beta(\alpha) = \chi_\beta(\alpha')$.*
3. *If $\beta' = \beta \cdot i^k$ with $k \in \mathbb{Z}$, $\chi_\beta(\alpha) = \chi_{\beta'}(\alpha)$.*
4. *(Quartic Reciprocity Law) If α, β are primary,*
$$\chi_\beta(\alpha) = \chi_\alpha(\beta) \cdot (-1)^{\frac{N(\alpha)-1}{4} \cdot \frac{N(\beta)-1}{4}}.$$
5. *(Complementary Laws) If $\beta = a + bi$ is primary,*
$$\chi_\beta(i) = i^{\frac{N(\beta)-1}{2}},\ \chi_\beta(1+i) = i^{\frac{a-b-b^2-1}{4}}.$$

One way to compute the quartic residue symbol, $\chi_\beta(\alpha)$, is by using the definition but this requires the knowledge of the prime factorization of β. In some cases (e.g., if it is not easy to factor β), it is more efficient to apply iteratively the properties of the quartic residue symbol as shown in Algorithm 1.

The first step of this algorithm is to use the modularity property and reduce α to an element α' equivalent to $\alpha \pmod{\beta}$. Then, using Proposition 4, α and β are replaced with primary elements α_1 and β_1. This is done in order to be able to apply iteratively the Quartic Reciprocity Law and the modularity property in a loop. The size of α and β decrease with each iteration of the loop and the algorithm stops when α becomes a unit. The algorithm will output $\chi_\beta(\alpha) = i^t$. The value t is initialized with 0 and it is updated each time the Complementary or the Quartic Reciprocity Laws are used.

Algorithm 1 Basic Algorithm Quartic Residuosity in $\mathbb{Z}[i]$

Input: $\alpha, \beta \in \mathbb{Z}[i] \setminus \{0\}$, $\gcd(\alpha, \beta) = 1$ and $(1+i) \nmid \beta$
Output: $c = \chi_\beta(\alpha)$ $(c = 0 \Leftrightarrow \chi_\beta(\alpha)$ is not defined)

$\alpha \leftarrow \alpha \bmod \beta$
if $\alpha = 0$ **then** $c = 0$
let primary $\alpha_1, \beta_1 \in \mathbb{Z}[i]$ be defined by
$\alpha = (i)^{i_1} \cdot (1+i)^{j_1} \cdot \alpha_1$ and
$\beta = (i)^{i_2} \cdot \beta_1$
let $m, n \in \mathbb{Z}$ be defined by $\beta_1 = m + ni$
$t \leftarrow \frac{m-n-n^2-1}{4} j_1 + \frac{m^2+n^2-1}{4} i_1 \bmod 4$
replace α with β_1, β with α_1
$t \leftarrow t + \frac{(N(\alpha)-1)(N(\beta)-1)}{8} \bmod 4$
while $N(\alpha) > 1$ **do**
  (LOOP INVARIANT: α, β are primary)
  let primary α_1 be defined by $\alpha \bmod \beta = (i)^{i_1} \cdot (1+i)^{j_1} \cdot \alpha_1$
  let $m, n \in \mathbb{Z}$ be defined by $\beta = m + ni$
  $t \leftarrow t + \frac{m-n-n^2-1}{4} j_1 + \frac{m^2+n^2-1}{4} i_1 \bmod 4$
  replace α with β, β with α_1
  $t \leftarrow t + \frac{(N(\alpha)-1)(N(\beta)-1)}{8} \bmod 4$
end while
if $N(\alpha) \neq 1$ **then** $c \leftarrow 0$ **else** $c \leftarrow i^t$

2.3 Characters of other orders

Let ω denote $\frac{-1+i\sqrt{3}}{2}$. It is the case that $\omega^2 + \omega + 1 = 0$. Characters of order 3 are defined on the ring of Eisenstein integers, the set $\mathbb{Z}[\omega] = \{a + b\omega \mid a, b \in \mathbb{Z}\}$. The Jacobi symbol can be extended for characters of order 3, similar to how it was defined for characters of order 4. Symbols of higher orders, possible when using integers of a cyclotomic field, are beyond the scope of this report.

2.4 Related Computational Problems

The security of the schemes presented in the following sections is based on the following problems:

Factorization. Given $n \in \mathbb{Z}$, find the prime factorization of n.

Let a *hard character* of order d on $\mathbb{Z}_n^*$ denote a non-trivial character and, for $d = 2$, a character other than the Jacobi symbol $(\frac{\cdot}{n})$.

MOVA^d. Let $n \in \mathbb{Z}$, t be a positive integer and χ be a *hard character* of order d on $\mathbb{Z}_n^*$. Given t pairs of the form $(x_i, \chi(x_i))$, with $1 \leq i \leq t$, and given $x \in \mathbb{Z}_n^*$, compute $\chi(x)$.

Depending on the value t of pairs used in an instance of a **MOVA**^{d} problem, there may be one or more characters that interpolate the t pairs

Prob. p	$F_1^{-1}(p)$	$F_2^{-1}(p)$
0.01	0.26974	0.00558
0.02	0.29341	0.0111
0.10	0.37851	0.05308
0.50	0.606	0.21172
0.90	0.90484	0.35899

Table 1: Distribution size for the largest two prime factors

$(x_i, \chi(x_i))$. The uniqueness of the character is given by the following theorem:

Theorem 1. *Let G and H be two finite Abelian groups. We denote by d the order of H. Let $x_1, x_2, \ldots, x_s \in G$ which span G'. The following properties are equivalent. In this case, we say that $x_1, x_2, \ldots, x_s$ H-generate G.*

- *For all $y_1, y_2, \ldots, y_s \in H$, there exists at most one group homomorphism $Hom : G \to H$ such that $Hom(x_i) = y_i$ for all $1 \leq i \leq s$.*
- *$G' + dG = G$.*

Proposition 6. *For d=2 or d=4 we have the following Karp reduction:*
$$MOVA^d \leq \textit{Factorization.}$$

Since solving an instance of the factorization problem would lead easily to a solution of the corresponding MOVA instance, we need to make sure that the instance of the factorization problem issued by our scheme is hard.

It is well known that for the particular case when $n = p \cdot q$, with p, q safe prime numbers, the factorization problem is hard. As the aim of our study is to construct cryptographic primitives without generating prime numbers, but by using randomly selected number, we must study the factorization problem in this context.

For n positive we write its unique prime decomposition as:
$$n = n_1 n_2 \cdots n_k, \text{ with } n_1 \geq n_2 \ldots \geq n_k.$$

The authors of [4] give theoretical results on the probability that, for a random number n, the k^{th} largest of its prime factors, n_k, is smaller than n^x where $0 < x < 1$. This is expressed by
$$F_k(x) = \lim_{N \to +\infty} \frac{P_k(x,N)}{N},$$
where $P_k(x, N)$ is the following function
$$P_k(x, N) = \#\{1 \leq n \leq N | n_k \leq N^x\}.$$

Table 1 in [6] illustrates some figures for finding the largest and second largest factor of a random number n.

It follows from [6] that in order to maintain the factorization hard for a number n picked at random, it is necessary to have the following inequality:

x	$F_2(x)$	l	k	$\|n\| = k \cdot l$
0.21172	0.5	1 276	45	57 420
0.01110	0.02	24 500	8	196 000
0.40681	0.96	660	750	495 000

Table 2: Required size of n for $L' = 1024$

$$\frac{C_{ECM}(x \cdot l)}{F_2(x)} \geq C_{GNFS}(L') \tag{1}$$

where $|n| = l$, $C_{ECM}(\lambda)$ denotes the complexity of the elliptic curve method to find a factor of length λ and $C_{GNFS}(\lambda)$ is the complexity of general number field sieve to factor an integer of length λ.

The inequality (1) denotes the following. To have good security guarantees on factoring a randomly picked number n, the complexity of finding with ECM a factor of size $x \cdot l$, taking into account the probability of having the second largest factor smaller than n^x, needs to stay close to the complexity of factoring a number of size L' bits. For instance, it is plausible to consider the factorization of n to be hard if the inequality holds for $L' = 1024$, e.g. a 1024 bits RSA modulus.

However, in the second scheme presented in this report, we do not take n as a random number. Instead, n is taken as a finite product of random numbers:

$$n = p_1 p_2 \cdots p_k, \text{ with } |p_i| = l \text{ for all } 1 \leq i \leq k, |n| = k \cdot l.$$

The description of security guarantees in this case is given by the following modification of (1):

$$min_{0<x<1}(C_{ECM}(x \cdot l), F_2(x)^{-k}) \geq C_{GNFS}(L').$$

A sufficient condition for that consists of taking x, y such that
$min[F_2(x)^{-k}, min_{x \leq u \leq y} \frac{C_{ECM}(ul)}{F_2(u)^k}, C_{ECM}(yl)] \geq C_{GNFS}(L')$.

For $x = y$ given, we can take l and k depending on L' such that

$$C_{ECM}(x \cdot l) = F_2(x)^{-k} = C_{GNFS}(L').$$

With different values for x we obtain an approximation for l and k that are illustrated in Table 2.

The primeless solution that we use, $n = p_1 p_2 \ldots p_k$ with $|p_i| = l$, requires to have a number n of hundreds of thousands of bits.

3 Probabilistic Encryption Scheme

The scheme illustrated by Algorithm 2, Algorithm 3 and Algorithm 4 is a probabilistic encryption scheme, introduced in [3]. Based on the properties of the quartic residue symbol, the sender is encrypting a bit b with the public key and the receiver is able to decrypt it using its secret key. We analyse here the security of this scheme.

Algorithm 2 Key generation

Input: Security parameter s.
Output: Public key: (n, p); Private key: γ.

1. Select a big $\gamma \in \mathbb{Z}[i]$, *i.e.* $\gamma = a' + b'i$, where $a', b' \in \mathbb{Z}$ and the size of a', b' depends on the security parameter s .
2. Compute $n = \gamma\bar{\gamma}$ $(n \in \mathbb{Z})$.
3. Pick a $p \in \mathbb{Z}$ such that $\chi_\gamma(p) = i$.

Algorithm 3 Encryption

Input: a bit b.
Output: the encryption c, $c \in \mathbb{Z}_n$.
Public key: (n, p).

1. Pick an $a \in \mathbb{Z}$ such that **if** $b = 0$ **then** $a \equiv 1 \pmod 4$ **else** $a \equiv 3 \pmod 4$.
2. $c \equiv p^a \pmod n$.

Algorithm 4 Decryption

Input: the encryption c, $c \in \mathbb{Z}_n$.
Output: a bit b.
Secret key: γ.

1. Compute the quartic residue symbol $\chi_\gamma(c)$ using Algorithm 1.
2. **if** $\chi_\gamma(c) = i$ **then** $b = 0$ **else** $b = 1$.

3.1 Correctness

Given this probabilistic scheme, first we need to prove its correctness. We formalize that as follows.By running the experiment $Exp^{correct}(s)$ (see Figure 1),

$$Prob[Exp^{correct}(s) \to 1] \text{ is } 1.$$

For this we can reuse a lemma from [3]:

Experiment $Exp^{correct}(s)$**:**

$((n,p),\gamma) \leftarrow$ Key Generation(1^s)
$b \leftarrow_U \{0,1\}$
$c \leftarrow$ Encryption$((n,p),b)$
$b' \leftarrow$ Decryption(γ,c)
Return $(b = b')$

Figure 1: Experiment defining the correctness of the probabilistic scheme.

Lemma 1. *[3]* $\chi_\gamma(c) = i^a$*, where* $c \equiv p^a \pmod n$ *and* (n,p) *is the public key computed by Algorithm 2.*

Proof. We have $c \equiv p^a \pmod n \Rightarrow$
$n|c - p^a$, where $n = \gamma\bar{\gamma} \Rightarrow$
$\gamma|c - p^a \Rightarrow c \equiv p^a \pmod \gamma$ (1)

Using the congruence (1), from the definition of a character and the way p is picked, we obtain that

$$\chi_\gamma(c) = \chi_\gamma(p^a) = \chi_\gamma(p)^a = i^a$$

□

Running the experiment $Exp^{correct}(s)$, a private key γ and a public key (n,p) are computed. The bit b, chosen randomly, is encrypted as $c \equiv p^a \pmod n$, where a is picked according to the value of b. Using Lemma 1, we have that $\chi_\gamma(c) = i^a$. If $a \equiv 1 \pmod 4$, then $\chi_\gamma(c_i) = i$ and $\chi_\gamma(c_i) = -i$ for $a \equiv 3 \pmod 4$.

From the way a is chosen in Algorithm 3 and provided that the receiver of the encrypted message knows the value of γ, with probability 1 the bit b is revealed correctly at the decryption process. Thus, $Prob[Exp^{correct}(s) \rightarrow 1]$ is 1 and the scheme is correct.

3.2 Security

In this subsection we present an algorithm which proves that the probabilistic encryption scheme illustrated above is not secure. A passive adversary $\mathcal{A}$ that runs Algorithm 5 is able to decrypt the values c that he intercepts with a non-negligible probability.

Notice that $\mathcal{A}$ does not know the value of γ and that only with n he cannot learn any information about an encrypted bit c by computing $\chi_n(c)$ (For $\chi_n(p) = -1$, we have $\chi_n(c) = \chi_n(p)^a = (-1)^a = -1$. The same reasoning can be applied for $\chi_n(p) = 1$.).

But we can see that if n has a gaussian factor $\gamma' \in \mathbb{Z}[i]$ (*i.e.* $\gamma'|n$) and $\chi_{\gamma'}(p) = \pm i$, then $\mathcal{A}$ is able to decrypt any message encrypted with the key (n,p). Indeed, if $\chi_{\gamma'}(p) = \pm i$, Lemma 1 is valid for γ'. $\mathcal{A}$ can apply

the same algorithm of decryption as the one described in Algorithm 4 for $\chi_{\gamma'}(p) = i$, where for $\chi_{\gamma'}(p) = -i$ he changes the condition from **if** $\chi_\gamma(c) = i$ to **if** $\chi_\gamma(c) = -i$. The number γ' can be considered an *equivalent* key to γ.

Based on this value γ', we introduce Algorithm 5, illustrated below.

Algorithm 5 Breaking the scheme

Input: the encrypted bit c, $c \in \mathbb{Z}_n$.
Output: a bit b.
Public key: (n, p).

1. Find 2 small factors of n, $f_1, f_2 \in \mathbb{Z}$, such that $f_j \equiv 1 \pmod 4$ for $1 \leq j \leq 2$. (See details below)
2. For f_1 and f_2 run *Cornacchia* and *Tonelli* algorithms to find $\gamma_j \in \mathbb{Z}[i]$ such that $\gamma_j \cdot \bar{\gamma_j} = f_j$, for $1 \leq j \leq 2$.
3. **if** $\chi_{\gamma_j}(p) = i$ or $\chi_{\gamma_j}(p) = -i$ **then** decrypt the value c using the following rules:
   for $\chi_{\gamma_j}(p) = i$: **if** $\chi_{\gamma_j}(c) = i$ **then** $b = 0$ **else** $b = 1$
   for $\chi_{\gamma_j}(p) = -i$: **if** $\chi_{\gamma_j}(c) = -i$ **then** $b = 0$ **else** $b = 1$
   otherwise abort.

In this algorithm, $\mathcal{A}$ tries to find two small prime factors of n by running a factorization algorithm, the complexity of which depends on the size of the smallest factor found, e.g., elliptic curve factorization. These factors can be of the form $f \equiv 1 \pmod 4$ or $f \equiv 3 \pmod 4$. As the quartic residue symbol is defined for any number $\alpha \in \{\alpha' \in \mathbb{Z}[i] \mid 1 + i \not| \alpha'\}$, then the value n in Algorithm 5 is odd so it does not have a power of 2 as factor. Also, recall that $n = a'^2 + b'^2$, $a', b' \in \mathbb{Z}$, therefore we can use the following characterization of it:

Theorem 2. *A positive integer n is a sum of two squares iff n is of the form $c \cdot d^2$ where c has no prime factor $f \equiv 3 \pmod 4$.*

From the characterization of the primes in $\mathbb{Z}[i]$, we know that given a prime $f \equiv 1 \pmod 4$, there exists a prime $\gamma \in \mathbb{Z}[i]$ such that $f = \gamma\bar{\gamma}$ and this value can be found with the Cornacchia and Tonneli algorithms. This number γ could be a good candidate for an equivalent key.

We have that $\chi_f(p) = \chi_\gamma(p) \cdot \chi_{\bar{\gamma}}(p)$. Since $f, p \in \mathbb{Z}$, $\mathcal{A}$ knows that $\chi_f(p) = \pm 1$ and with probability $\frac{1}{2}$, $\chi_\gamma(p)$ and $\chi_{\bar{\gamma}}(p)$ are $\pm i$. Thus, $\mathcal{A}$ needs 2 small factors of the form $f \equiv 1 \pmod 4$ in order to be sure that he finds, with probability $\frac{3}{4}$, an equivalent key γ' such that $\gamma' | n$ and $\chi_{\gamma'}(p) = \pm i$.

Prime factors of the form $f \equiv 3 \pmod 4$ are also primes in $\mathbb{Z}[i]$ and they cannot be further used to find γ with $\gamma\bar{\gamma} = f^{2k}$, $k \in \mathbb{Z}$, since $\mathbb{Z}[i]$ is a unique factorization domain.

If we analyse the complexity of Algorithm 5, we can see that the most costly operations are the factorization and the Cornacchia & Tonelli algo-

rithms. The factorization will run in polynomial time with respect to s as we require to find small factors, e.g. 5 or 13. Cornacchia and Tonelli algorithms also run in polynomial time with respect to the size of f. Provided that n has two small factors of the required form, the total complexity of the algorithm remains polynomial. The probability of this event is expressed by

$$Prob_{n\in\mathbb{Z},2\not|n}[f_1 \equiv 1 \pmod 4, f_2 \equiv 1 \pmod 4, f_1 \neq f_2\ ] = \frac{1}{2^{2}.f_1.f_2},$$

which is non-negligible.

Whenever n has two small prime factors f such that $f \equiv 1 \pmod 4$, the probability of success is $\frac{3}{4}$.

Thus, the scheme is insecure (i.e. not even one-way encryption secure).

Our goal is to be able to use n which has *some* large factors, rather than n which has *no* small factor. So, we have to change the scheme.

4 New Proposal

In this section we propose an alternative to the first primeless scheme (i.e. it does not generate primes). We design a key agreement protocol that can be easily transformed in an encryption scheme, without loss of the security guaranty. We discuss the extension of the scheme to characters of order 4, making it more efficient.

4.1 Key Agreement

In the key agreement scenario we have two participants: A (Alice) and B (Bob) who want to agree on a secret key. In our scheme, A and B agree on a bit value, K, but by running the protocol several times in parallel, we obtain a bitstring key agreement. The agreement scheme is graphically illustrated in Figure 2.

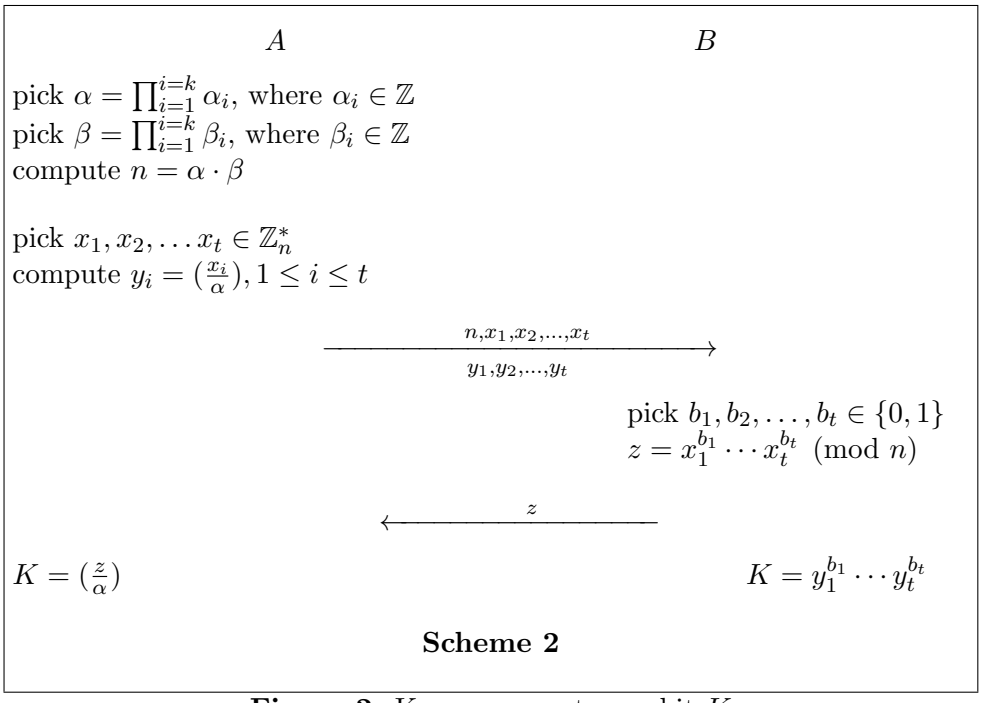

Figure 2: Key agreement on a bit K

In the setup phase, A is choosing randomly a list of large integers α_i, β_i and computes $n = \alpha \cdot \beta$. The value n is made public and the value α is kept secret. Jacobi symbol $(\frac{\cdot}{\alpha})$ is a hard character of order 2. Then, A picks randomly t values, $x_1, x_2, \ldots, x_t$ from $\mathbb{Z}_n^*$ and computes, using the Jacobi symbol $(\frac{\cdot}{\alpha})$, $y_i = (\frac{x_i}{\alpha})$ for all $1 \leq i \leq t$. If all $y_i's$ are equal then A drops all the $x_i's$ and restarts by choosing other values. Participant B receives the x_i's, y_i's and the value n. He picks randomly t bits, $b_1, b_2, \ldots, b_t$, computes

$z = x_1^{b_1} \cdots x_t^{b_t}$ and sends this value to A. Again we require for $b_i's$ not to be all equal to 0.

Finally, the two participants can agree on a bit value K, where
$K = (\frac{z}{\alpha}) = (\frac{x_1^{b_1} \cdots x_t^{b_t}}{\alpha}) = (\frac{x_1}{\alpha})^{b_1} \cdots (\frac{x_t}{\alpha})^{b_t} = y_1^{b_1} \cdots y_t^{b_t}$.

Choice of parameters

Let $s \in \mathbb{Z}$ be the security parameter. For the setup phase we require for β_i and α_i to be of size l for all $1 \leq i \leq t$. In the Background section we looked at how the size of n varies with the size of l and the number k. We assume that these values are large enough for the complete factorization to be hard. In this key agreement we don't consider α as a long term key as, at each run of the scheme, we can change α and n.

We pick value t such that, by Theorem 1, we obtain the uniqueness of the homomorphism. Also we require to have $t > s$ so that for an adversary $\mathcal{A}$ to be hard to try all 2^t combinations of b_i's.

Complexity

In the setup phase, A generates randomly $2k$ numbers and performs $2k-1$ multiplications to obtain value n. These operations are performed in $O(2kl + l^2(2k-1))$.

The choice of $x_i's$ from $\mathbb{Z}_n^*$ and the computation of $y_i's$ and K are done in $O(2tkl + t(2kl)^2)$. Thus, the total computation done by participant A is of order $O(tk^2l^2)$.

Participant B picks t bits and performs at most $t-1$ multiplications to compute z, which takes $O(t(2kl)^2)$. Value K is computed within a complexity of order $O(t)$ as $y_i's \in \{+1, -1\}$ and $b_i's \in \{0, 1\}$, obtaining a total complexity of order $O(tk^2l^2)$.

As it was shown in Background section, the total complexity varies with respect to l. For a big value l, e.g more than 10 000, value k is small and the complexity is of order $O(tl^2)$. On the other hand, for a smaller value l, e.g a few hundreds, k is comparable with l, and Scheme 2 has a complexity of order $O(tl^4)$.

Corectness

For the two participants of the key agreement, A and B, it is necessary to have the following property: they agree on the same value K.

Given that both participants are honest, we have the following lemma;

Lemma 2. *$(\frac{z}{\alpha}) = y_1^{b_1} y_2^{b_2} \cdots y_t^{b_t}$, where the values $z, \alpha, y_i's$, $b_i's$ are computed as in Scheme 2.*

Proof. We have $z = x_1^{b_1} \cdots x_t^{b_t} \pmod{n}$ with $\alpha / n \Rightarrow$

$$z = x_1^{b_1} \cdots x_t^{b_t} \pmod{\alpha}$$

Using Proposition 1, we obtain:

$(\frac{z}{\alpha}) = (\frac{x_1^{b_1} \cdots x_t^{b_t}}{\alpha}) = (\frac{x_1^{b_1}}{\alpha}) \cdots (\frac{x_t^{b_t}}{\alpha}) = (\frac{x_1}{\alpha})^{b_1} \cdots (\frac{x_t}{\alpha})^{b_t} = y_1^{b_1} y_2^{b_2} \cdots y_t^{b_t}$

□

Thus, Scheme 2 is correct and in the end both A and B will share the same secret value K.

Security

Here we analyse the security of the new proposed key agreement. In this context, we can see that a passive adversary $\mathcal{A}$ observes the messages sent by A and B and in the end he can try to guess bit K. Thus, $\mathcal{A}$ has access to t pairs $(x_i, (\frac{x_i}{\alpha}))$ and value n, which corresponds to an instance of the MOVA[2] problem. We say that the key agreement Scheme 2 is secure if, after the run of it,

$$Adv_{\mathcal{A}}(s) = Prob_{x,b}[K = K'] - \tfrac{1}{2},$$

is negligible in terms of s, where K' is the bit guessed by $\mathcal{A}$, K is the agreed key , x and b denote the values $x_i's$ and $b_i's$ that are generated during the run of the key agreement.

For our security proof, we use the following theorem from [7] :

Theorem 3. *Let $\varphi : G \to \mathbb{Z}_d$ be a group homomorphism. If one can compute a f such that $Prob_{z \in G}(f(z) \neq \varphi(z)) \leq \frac{\xi}{12}$ [1] with a constant $\xi < 1$, then one can compute φ in a number of calls to f bounded by a polynomial in $log(\#G)$.*

This theorem states that if one is able to approximate φ with a small probability of error, then one must know the homomorphism φ. We want to apply this to our proof as it would reduce the security to the MOVA^d problem. If for an adversary $\mathcal{A}$, $Adv_{\mathcal{A}}(s)$ is non-negligible, then according to Theorem 2, $\mathcal{A}$ is able to compute φ and thus solve the MOVA^d problem.

For the proposed scheme, we take φ to be the Jacobi symbol $(\frac{\cdot}{\alpha})$: $\mathbb{Z}_n^* \to \{+1, -1\}$. Any adversary can build function $f(x)$ as follows: with the $x_i's$ and $y_i's$ from the MOVA^d problem, he picks randomly a number of p combinations of bits $b_1', b_2', \ldots b_t' \in \{0, 1\}$ and computes $z' = x \cdot x_1^{b_1'} \cdots x_t^{b_t'} \pmod{n}$ and $K' = (\frac{x}{\alpha}) \cdot y_1^{b_1'} \cdots y_t^{b_t'}$ from which he deduces $(\frac{x}{\alpha})$. The value p is bounded by a polynomial in $\log(\#\mathbb{Z}_n^*)$ as it is required in Theorem 3.

In order to apply Theorem 3 we only need to prove that the values $z = x_1^{b_1} \cdots x_t^{b_t} \pmod{n}$ generate uniformly $\mathbb{Z}_n^*$. There is a similar result presented in [5] for the generalized compact knapsack. We conjecture that value z has a distribution that is indistinguishable from the uniform distribution.

Thus, assuming that the Factorization and MOVA^d problems are hard we obtain that Scheme 2 is secure, result expressed by Corollary 1.

[1] $\frac{\xi}{12}$ can be replaced with $1 - \frac{1}{d}$ [7].

Corollary 1. *Assuming that MOVA^d is hard, then no adversary $\mathcal{A}$ can guess the bit K from one run of the agreement but with a probability bounded by $\frac{1}{2}$.*

4.2 Probabilistic Encryption Scheme

In this subsection we show how we can turn the key agreement Scheme 2 into an encryption scheme. With the small changes that we bring, we can maintain the security. The new encryption scheme is illustrated bellow in Figure 3.

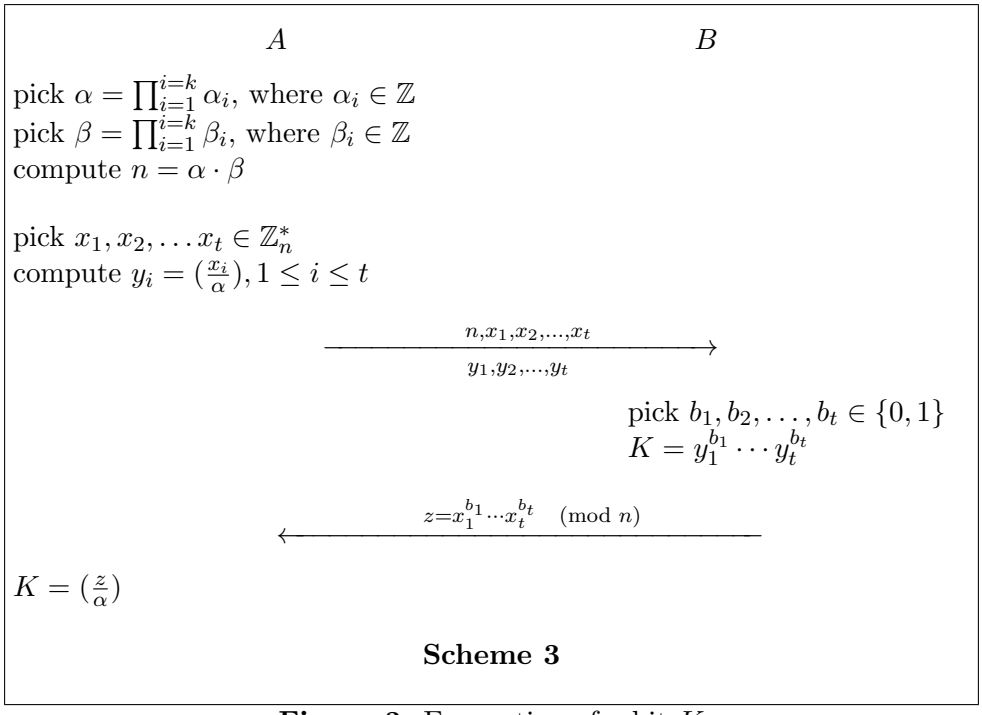

Figure 3: Encryption of a bit K

Participant A proceeds with the same steps as before. He picks value n, α and sends to B the values $x_i's$, $y_i's$ and n. The only difference appears in the choices that participant B makes. He first selects an index i for a pair (x_i, y_i), with $y_i = -1$, from what he receives. With the rest of the values, he selects randomly $t-1$ bits, $b_1, \ldots, b_{i-1}, b_{i+1}, \ldots, b_{t-1}$, selects b_i such that $\prod y_j^{b_j} = K = (-1)^b$ and computes $z = x_1^{b_1} \cdots x_t^{b_t} \pmod{n}$.

Then, he sends z to be the ciphertext for K. Knowing value α, A is able to decrypt z and to obtain plaintext bit K.

Corectness

Because in Scheme 3 value K is computed in the same manner as in Scheme 2, we can use Lemma 2. The encryption scheme is correct and participant A decrypts correctly z, provided that he knows the secret value α.

Security

Concerning the security of this encryption scheme we can use the same results as for the key agreement as the goal of the adversary is the same: to guess whether K is 0 or 1.

Corollary 2. *Assuming that MOVA^d is hard, then no adversary $\mathcal{A}$ can guess the bit K from the run of the encryption scheme but with a probability bounded by $\frac{1}{2}$ and the scheme is IND-CPA secure.*

4.3 Using characters of order 4

It is possible to improve the scheme presented in this section by using characters of order 4. With the quartic residue symbol, the two participants agree on a 2 bit value K (or encrypt two bits) with each run of the protocol. In this scenario, participant A is choosing β_i and α_i to be Gaussian integers and computes n as $\prod_i \alpha_i \cdot \bar{\alpha_i} \cdot \beta_i \cdot \bar{\beta_i}$. B is now choosing the $b_i's$ from $\{0, 1, 2, 3\}$. The correctness and the security proof of the scheme are maintained.

5 Conclusions

In this project we have introduced a key agreement and an encryption scheme that uses primeless cryptography. It was shown that by using only random numbers and the notion of a character, it is possible to build cryptographic primitives that are proven to be secure. As we are not generating prime numbers, we compute a value n as a product of random numbers. We use this approach in order to maintain the hardness of factorization. Because of that, the size of n is of hundreds of thousands of bits.

One improvement, when using primeless cryptography, is the fact that the complexity of generating value n, in the setup, is lower than the complexity of generating prime numbers. On the other hand, randomness has in this case drawbacks. In practice it may be difficult to work with big values as n. Also, once we use in a primitive these big numbers, the complexity of every operation (multiplication, addition, exponentiation) is higher. This may lead to a total complexity, for a primitive that uses primeless cryptography, bigger than one that uses prime numbers.

As a future work we need to do a more detailed analysis on the required size of l and k. With such a result we can obtain the size of n, not only some approximations.

An extension of this project might be the following: construct a cryptographic primitive, using characters of order 4, that is more efficient than the schemes presented in here. So far, he are able to send or encrypt only 2 bits at each run.

Bibliography

[1] Henri Cohen. *A course in computational algebraic number theory.* Springer-Verlag New York, Inc., New York, USA, 1993.

[2] K. Ireland and M. Rosen. A classical introduction to modern number theory. 1990.

[3] Ehsan Kazemi. Primeless cryptography. Semester project, EPFL, January 2011.

[4] Donald E. Knuth and Luis Trabb Pardo. Analysis of a simple factorization algorithm, 1976.

[5] Daniele Micciancio. Generalized compact knapsacks, cyclic lattices, and efficient one-way functions. *Computational Complexity*, 16(4):365–411, 2007.

[6] Jean Monnerat. *Short undeniable signatures:Design, Analysis, and Applications.* PhD thesis, École Polytechnique Fédérale de Lausanne, 2006.

[7] Jean Monnerat and Serge Vaudenay. Undeniable signatures based on characters: How to sign with one bit. In *Public Key Cryptography*, pages 69–85, 2004.