A New Public-Key Cryptosystem via Mersenne Numbers

Divesh Aggarwal* Antoine Joux† Anupam Prakash‡ Miklos Santha§

May 30, 2017

Abstract

In this work, we propose a new public-key cryptosystem whose security is based on the computational intractability of the following problem: Given a Mersenne number $p = 2^n - 1$, where n is a prime, a positive integer h, and an n-bit integer H, decide whether their exist n-bit integers F, G each of Hamming weight less than h such that $H = \frac{F}{G}$ modulo p.

1 Introduction

1.1 Motivation

Since the seminal work of Diffie and Hellman [DH76] which presented the fundamentals of public-key cryptography, one of the most important goal of cryptographers has been to construct secure and practically efficient public-key cryptosystems. Rivest, Shamir, and Adleman [RSA78] came up with the first practical public-key cryptosystem based on the hardness of factoring integers, and it remains the most popular scheme till date.

Shor [Sho97] gave a quantum algorithm that solves the abelian hidden subgroup problem and as a result solves both discrete logarithms and factoring. Back in 1994, this was not considered a real threat to the practical cryptographic schemes since quantum computers were far from being a reality. However, given the recent advances in quantum computing, there is serious effort in both the industry and the scientific community to make information security systems resistant to quantum computing. In fact, the National Institute of Standards and Technology (NIST) is now beginning to prepare for the transition into quantum-resistant cryptography and has announced a project where they are accepting submissions for quantum-resistant public-key cryptographic algorithms [NIS17].

In the recent years, some presumably quantum-safe public-key cryptosystems have been proposed in the literature. Perhaps the most promising among these are those based on the hardness

*School of Computing and CQT, NUS.

†Chaire de Cryptologie de la Fondation de l'UPMC; Sorbonne Universités, UPMC Univ Paris 06, CNRS, LIP6 UMR 7606, Paris, France

‡School of Physical and Mathematical Sciences, Nanyang Technological University and Centre for Quantum Technologies, National University of Singapore, Singapore.

§IRIF, Université Paris Diderot, CNRS, 75205 Paris, France; and Centre for Quantum Technologies, National University of Singapore Singapore 117543

of lattice problems like Learning with Errors (LWE) based cryptosystems [Reg09], Ring-LWE based cryptosystems [LPR10] and NTRU [HPS98]. While these cryptosystems have so far resisted any classical or quantum attacks, it cannot be excluded that such attacks are possible in the future. In fact, there have been some, albeit unsuccessful, attempts at a quantum algorithm solving the LWE problem [ES16]. In particular, there is no unifying complexity-theoretic assumption (like NP-hardness) that relates the difficulty of breaking all these cryptosystems. Thus, it is desirable to come up with promising new proposals for public-key cryptosystems.

It is worthwhile to note that even though the concept of public-key cryptography was introduced four decades ago, the number of existing public-key cryptographic schemes whose hardness does not depend on the hardness of factoring or finding short vectors in lattices is not very large [KLC+00, McE78, LvTMW09, GWO+13, NS97]. This is not an exhaustive list but it illustrates the various approaches that have been tried. The rarity of proposals for potentially quantum safe public key cryptosystems further motivates the problem of constructing such cryptosystems.

1.2 Our Cryptosystem

Our cryptosystem is based on arithmetic modulo so called Mersenne numbers, i.e., numbers of the form $p = 2^n - 1$, where n is a prime. These numbers have an extremely useful property: For any number x modulo p, and $y = 2^z$, where z is a positive integer, $x \cdot y$ is a cyclic shift of x by z positions and thus the Hamming weight of x is unchanged under multiplication by powers of 2. Our encryption scheme is based on the simple observation that when we consider $H = \frac{F}{G} \pmod p$, where the binary representation of F and G modulo p has low Hamming weight, then H looks pseudorandom, i.e., it is hard to distinguish H from a random integer modulo p. In order to encrypt a bit $b \in \{0, 1\}$, the encryption algorithm chooses two random numbers A, B modulo p of low Hamming weight and then outputs

$$C := (-1)^b \cdot (A \cdot H + B) ,$$

where H is the public key, and G is the private key. Given the private key, one can compute $C \cdot G$ to decrypt the bit b by checking whether $C \cdot G$ has low Hamming weight (corresponding to $b = 0$) or high Hamming weight (corresponding to $b = 1$). For more details on our scheme and the underlying security assumption, we refer the reader to Section 3 and 4.

Our cryptosystem is somewhat similar to the NTRU cryptosystem [HPS98] , in the sense that NTRU also uses the idea that if f, g are elements of a certain ring R, g is invertible, and both f and g are short with respect to some metric, then f/g is pseudorandom. However, the NTRU cryptosystem requires arithmetic operations over a ring $\mathbb{Z}_q[x]/(x^n - 1)$, where q is a large integer. Contrary to this, our cryptosystem requires only binary operations and can be made significantly more efficient. Moreover, Bos et al [BKLM11] showed how to enhance the efficiency of arithmetic operations modulo Mersenne primes.

In Section 5.1, we mention some possible approaches to attack our cryptosystem, and in Section 5.2 we discuss active attacks. In section 6, we propose a method to make our scheme CCA-secure.

2 Preliminaries

Notations. The Hamming weight of an n-bit string s is the total number of 1's in s and is denoted by $\mathsf{Ham}(s)$. By $x \pmod{2^n-1}$, we denote the number $y \in \{0,1,\ldots,2^n-2\}$ such that $x \equiv y \pmod{2^n-1}$. We use the binary representation of x and the corresponding integer $x \in \{0,1,\ldots,2^n-2\}$ interchangeably, and it will be clear from the context which of the two we are talking about.

For any distinguisher D that outputs a bit $b \in \{0,1\}$, the distinguishing advantage to distinguish between two random variables X and Y is defined as:

$$\Delta^D(X\ ;\ Y) := |\Pr[D(X)=1] - \Pr[D(Y)=1]|\ .$$

The following lemma is well known and easy to see.

Lemma 1. *Given a probabilistic polynomial time computable function f on two random variables X and Y, if there is a probabilistic polynomial time distinguisher D that distinguishes between $f(X)$ and $f(Y)$ with advantage δ, then there is a probabilistic polynomial time distinguisher D' that distinguishes between X, and Y with advantage δ.*

2.1 Mersenne Numbers and Mersenne Primes

A Mersenne number p is a number of the form 2^n-1 where n is a prime. If 2^n-1 is itself a prime number, then it is called a Mersenne prime. Note that if n is a composite number of the form $n = k\ell$, then 2^k-1 and $2^\ell-1$ divide p, and hence p is not a prime. The smallest Mersenne primes are

$$2^2-1, 2^3-1, 2^5-1, 2^7-1, 2^{13}-1, 2^{17}-1, \ldots$$

Lemma 2. *Let $A, B \in \{0,1\}^n$ and $p = 2^n-1$. Then*

1. $\mathsf{Ham}(A+B \pmod p) \le \mathsf{Ham}(A) + \mathsf{Ham}(B)$.
2. $\mathsf{Ham}(A \cdot B \pmod p) \le \mathsf{Ham}(A) \cdot \mathsf{Ham}(B)$.
3. *If $A \neq 0^n$,* $\mathsf{Ham}(-A \pmod p) = n - \mathsf{Ham}(A)$.

Proof. 1. Without loss of generality, we assume $A \neq 1^n$, since the result is obviously true if $A = 1^n$. We prove the result by induction on the Hamming weight of B. We first assume that $\mathsf{Ham}(B) = 1$, and let i be the index on which B takes the value 1. Since addition modulo p is invariant by rotation, we may assume that $i = 0$ and thus $B = 1$. In this case, if A is even, then $A+B$ is just a copy of A with the low order bit set to 1 and the Hamming weight increases by 1. Otherwise, A ends by a sequence of the form $011\ldots1$ and after adding 1, it becomes $100\ldots0$. In that case, the Hamming weight of $A+B$ is at most the Hamming weight of A.

Let $\mathsf{Ham}(B) = k$. Then $B = B_1 + B_2$, where each B_1 is an n-bit string with Hamming weight $k-1$, and B_2 is a string of Hamming weight 1. By the induction hypothesis:

$$\mathsf{Ham}(A+B \pmod p) = \mathsf{Ham}((A+B_1)+B_2) \le \mathsf{Ham}(A+B_1)+1 \le \mathsf{Ham}(A)+(k-1)+1,$$

and the result follows.

2. We can write $B = B_1 + \cdots + B_k$, where each $B_i \in \{0,1\}^n$ has Hamming weight 1. Then $A \cdot B_i \pmod p$ is A_i obtained by shifting A upto rotation by i_j positions to the left, where $B_i = 2^{i_j}$. Thus $\mathsf{Ham}(A_i) = \mathsf{Ham}(A)$, and

$$A \cdot B \pmod p = A_1 + \cdots + A_i \pmod{2^n - 1}.$$

   The result then follows from part (1).

3. Notice that 1^n is the binary representation of p, and hence $-A \pmod p = p - A$ is the bitstring obtained by replacing 0's by 1's and 1's by 0's in A.

□

2.2 Security Definitions

In the following, we give the most widely used security definitions for public-key cryptography. Let $(\mathsf{Enc}, \mathsf{Dec})$ be the given public-key encryption scheme with public-key, and private-key being pk and sk, and let $|C|$ denote the length of the ciphertext C. We denote the security parameter by λ.

Definition 1. The public-key encryption scheme $(\mathsf{Enc}, \mathsf{Dec})$ is said to be *semantically secure* if for any probabilistic polynomial time distinguisher and any pair of messages m_0, m_1 of equal length, given the public key pk, the advantage for distinguishing $C_0 = \mathsf{Enc}(\mathsf{pk}, m_0)$ and $C_1 = \mathsf{Enc}(\mathsf{pk}, m_1)$ is at most $\frac{\mathrm{poly}(|C_i|)}{2^\lambda}$.

Definition 2. The public-key encryption scheme $(\mathsf{Enc}, \mathsf{Dec})$ is said to be *secure under chosen ciphertext attacks* if for any probabilistic polynomial time distinguisher that is given access to an oracle that decrypts any given ciphertext, the following holds: For any pair of messages m_0, m_1 of equal length, given the public key pk, the advantage for distinguishing $C_0 = \mathsf{Enc}(\mathsf{pk}, m_0)$ and $C_1 = \mathsf{Enc}(\mathsf{pk}, m_1)$ is at most $\frac{\mathrm{poly}(|C_i|)}{2^\lambda}$ under the assumption that the distinguisher does not query the oracle with C_0 or C_1.

3 Basic bit-by-bit Encryption

In the following, we describe the basic scheme to encrypt a single bit $b \in \{0,1\}$.

Key Generation.

- Given the security parameter λ, choose a Mersenne prime $p = 2^n - 1$ and an integer h such that $\binom{n-1}{h-1} \geq 2^\lambda$ and $4h^2 < n$. For a concrete choice of parameters, refer to Section 6.1.
- Choose F, G to be two independent n-bit strings chosen uniformly at random from all n-bit strings of Hamming weight h.
- Set $\mathsf{pk} := H = \frac{F}{G} \pmod p$, and $\mathsf{sk} := G$.

Encryption. The encryption algorithm chooses two independent strings A, B uniformly at random from all strings with Hamming weight h. A bit b is encrypted as

$$C = \mathsf{Enc}(\mathsf{pk}, b) := (-1)^b \, (A \cdot H + B) \pmod{p} \, .$$

Decryption. The decryption algorithm computes $d = \mathsf{Ham}(C \cdot G \pmod{p})$. If $d \leq 2h^2$, then output 0; if $d \geq n - 2h^2$, then output 1. Else output $\perp$.

The correctness of the decryption is immediate from Lemma 2. To see this, note that $C \cdot G \pmod{p} = (-1)^b \cdot (A \cdot F + B \cdot G) \pmod{p}$ which, by Lemma 2 has Hamming weight at most $2h^2$ if $b = 0$, and at least $n - 2h^2$ if $b = 1$.

4 Semantic Security of the bit-by-bit Scheme

For proving semantic security, we need the following assumptions.

Definition 3.

- The *Mersenne Low Hamming Combination Assumption* states that given an n-bit Mersenne prime $p = 2^n - 1$, and an integer h, the advantage of any probabilistic polynomial time adversary attempting to distinguish between $(R_1, AR_1 + B)$ and (R_1, R_2) is at most $\frac{\mathrm{poly}(n)}{2^\lambda}$, where R_1, R_2 are uniformly random n-bit strings, and A, B, are independently chosen n-bit strings each having Hamming weight h.
- The *Mersenne Low Hamming Ratio Assumption* states that given an n-bit Mersenne prime $p = 2^n - 1$, and an integer h, the advantage of any probabilistic polynomial time adversary attempting to distinguish between $\frac{F}{G} \pmod{p}$ and R is at most $\frac{\mathrm{poly}(n)}{2^\lambda}$, where R is a uniformly random n-bit strings, and F, G, are independently chosen n-bit strings each having Hamming weight h.

Before proving the security, we show the following:

Lemma 3. *If the two assumptions in Definition 3 hold, then the advantage of any probabilistic polynomial time adversary attempting to distinguish between (H, C^*) and (H, R_2) is at most $\frac{\mathrm{poly}(n)}{2^\lambda}$, where $H = \frac{F}{G} \pmod{p}$, $C^* = AH + B \pmod{p}$, A, B, F, G, are independently chosen n-bit strings each having Hamming weight h and R_2 is a uniformly random n-bit string.*

Proof. By the triangle inequality, we have that for any distinguisher D,

$$\begin{aligned}\Delta^D((H, C^*) \, ; \, (H, R_2)) \leq \Delta^D((H, AH + B) \, ; \, (R_1, AR_1 + B)) \\ + \Delta^D((R_1, AR_1 + B) \, ; \, (R_1, R_2)) + \Delta^D((R_1, R_2) \, ; \, (H, R_2)) \, .\end{aligned}$$

Now, from Lemma 1 and the second assumption of Definition 3, we have that there exists a distinguisher D' such that

$$\Delta^D((H, AH + B) \, ; \, (R_1, AR_1 + B)) = \Delta^{D'}(H \, ; \, R_1) \leq \frac{\mathrm{poly}(n)}{2^\lambda} \, .$$

Similarly, using both assumptions from Definition 3 and Lemma 1,

$$\Delta^D((R_1, AR_1 + B)\ ;\ (R_1, R_2)) \leq \frac{\text{poly}(n)}{2^\lambda}$$

and

$$\Delta^D((R_1, R_2)\ ;\ (H, R_2)) \leq \frac{\text{poly}(n)}{2^\lambda}\ ,$$

thereby implying the result. □

Theorem 1. *The bit-by-bit scheme* $(\mathsf{Enc}, \mathsf{Dec})$ *is semantically secure under the Mersenne Low Hamming Combination Assumption.*

Proof. Let k be any integer. Consider the following to be chosen independently and uniformly from all n-bit strings of Hamming weight h

$$F, G, A_1, B_1, A_2, B_2, \ldots, A_k, B_k\ .$$

Let $H = \frac{F}{G} \pmod p$, and let $C_i^* = A_i H + B_i \pmod p$. Also, let $R_1, R_2, \ldots, R_k$ be chosen uniformly at random. Define the random variable X_i for $i = 0, \ldots, k$ as follows:

$$X_i := H, C_1^*, C_2^*, \ldots, C_i^*, R_{i+1}, \ldots, R_k\ .$$

Consider a probabilistic polynomial time distinguisher D, let $\Delta^D(X_0, X_k) = \delta$. By the triangle inequality, there exists $i \in \{0, \ldots, k-1\}$ such that $\Delta^D(X_i, X_{i+1}) \geq \frac{\delta}{k}$. Notice that X_i (respectively, X_{i+1}) is $f(H, C_i^*)$ (respectively, $f(H, R_i)$) where f is a randomized function obtained by sampling $A_1, B_1, \ldots, A_{i-1}, B_{i-1}$ independently and uniformly from all n-bit strings of Hamming weight h, computing $C_j^* = A_j H + B_j$ for $1 \leq j \leq i-1$, and sampling $R_{i+1}, \ldots, R_k$ uniformly and independently from all bistrings of length n. Thus, using Lemma 1, we have that there is an efficient distinguisher D' such that

$$\Delta^{D'}((H, C_i^*)\ ;\ (H, R)) \geq \frac{\delta}{k}\ .$$

Thus, from Lemma 3, we have that

$$\Delta^D(X_0\ ;\ X_k) = \delta \leq \frac{\text{poly}(k, n)}{2^\lambda}\ .$$

For any k-bit message $m = b_1, \ldots, b_k$, where $b_1, \ldots, b_k \in \{0, 1\}$, the (public-key, ciphertext) pair is distributed as

$$H, (-1)^{b_1} C_1^*, \ldots, (-1)^{b_k} C_k^*\ ,$$

which by the argument above and Lemma 1 cannot be distinguished from

$$Y^{(m)} := H, (-1)^{b_1} R_1, \ldots, (-1)^{b_k} R_k\ ,$$

with advantage more than $\frac{\text{poly}(k,n)}{2^\lambda}$.

Note that for any two k-bit messages m_0, m_1, $Y^{(m_0)}$ and $Y^{(m_1)}$ are identically distributed. Thus, for any probabilistic polynomial time distinguisher D,

$$\Delta^D(\mathsf{Enc}(\mathsf{pk}, m_0), \mathsf{Enc}(\mathsf{pk}, m_1)) \leq \frac{\text{poly}(k, n)}{2^\lambda}\ ,$$

which proves the desired result. □

5 Analysis of our Security Assumption

5.1 Attempts at Cryptanalysis

In this section, we mention some of the approaches that we tried to break our scheme and thereby mention the conjectured security guarantee.

For cryptanalysis, it is often more convenient to talk about search problems. Thus, we introduce the following search problem whose solution would imply an attack on our cryptosystem.

Definition 4 (Mersenne Low Hamming Ratio Search Problem)**.** *Given an n-bit Mersenne number $p = 2^n - 1$, an n-bit string H, and an integer h, find two n-bit strings F, G, each of Hamming weight at most h such that $H = \frac{F}{G} \pmod p$.*

For the remainder of the paper, we call this problem $\mathcal{P}$. It is easy to see that if one can efficiently solve the problem $\mathcal{P}$, then one can break the assumption in Definition 3, and hence the security of our cryptosystem. It is thus important to study the hardness of this problem.

Hamming Weight Distribution. Let Y be an n bit string generated as $Y = F/G$ where F, G are chosen uniformly at random from n bit strings with Hamming weight h. A basic test for the assumption that Y is pseurandom is to check that the distribution of $\mathsf{Ham}(Y)$ is close the distribution of the Hamming weights for a uniformly random n bit string.

If X is a uniformly random n bit string, the random variable $f(X) = \frac{\mathsf{Ham}(X) - n/2}{\sqrt{n/4}}$ is approximated by the standard normal random variable $N(0, 1)$. We generated several samples $Y = F/G$ where F, G are uniformly distributed over strings of Hamming weight $\sqrt{n}$. A quantile-quantile plot of $f(Y_i)$ against samples from $N(0, 1)$ is close to a straight line and does not show significant deviations from normality.

Of course, one could also perform more advanced statistical tests, such as the NIST suite [RSN+01]. However, in the context of cryptographic schemes, such tests only serve as sanity checks and it is preferable to focus on dedicated cryptanalysis.

Lattice-based Attacks. The analogy with NTRU suggests that one could hope to come up with lattice reduction based cryptanalytic attacks on our scheme. A possible approach to solve problem $\mathcal{P}$ stated in Definition 4 is to consider the $(2n + 1)$ dimensional lattice $L(B)$ with basis given by the columns of the matrix below.

$$B = \begin{bmatrix} K.H & K.(2H \bmod p) & \dots & K.(2^{n-1}H \bmod p) & K.1 & K.2 & \dots & K.2^{n-1} & K.p \\ 1 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 1 & 0 \end{bmatrix}$$

The columns of B are linearly independent so the lattice $L(B)$ has full rank and all the entries of B are n bit integers. The integer K is chosen to be sufficiently large so that any short vector in $L(B)$ has first coordinate 0.

If there exist n bit numbers F, G such that $H = F/G$ then $B(G, -F, 0)^T = (0, G, -F)^T$. If there exist F, G with $\mathsf{Ham}(F) = \mathsf{Ham}(G) = h$ and $H = F/G$ then there is a vector in $L(B)$ with norm $\sqrt{2h}$. As a consequence, one could hope to perform lattice reduction algorithm to find this short vector (or a rotated copy) and thus recover a solution to the problem $\mathcal{P}$.

However, there are many vectors in the lattice which are even shorter than this one. For example, we know that for all i there exists an integer α such that:

$$2\,(K.((2^i H \pmod p))) - (K.(2^{i+1} H \pmod p)) - \alpha\,(K.p) = 0.$$

Thus, $B(0^{i-1}, 2, -1, 0^{2n-i-1}, \alpha)^T = (0, 0^{i-1}, 2, -1, 0^{2n-i-1})^T$, and we have found a vector of norm $\sqrt{5}$ in the lattice. These very short vectors prevent usual lattice reduction algorithms to find the vector that corresponds to the solution of $\mathcal{P}$. To get rid of these parasitical short vectors, one could perform lattice reduction in the L_∞ norm instead of the Euclidean norm, however, we do not know efficient algorithms for this purpose.

Meet in the middle attacks. The problem $\mathcal{P}$ can also be reformulated as a problem similar to the cyclic subset sum problem. Let v_i be the n-bit vector that is the binary representation of $2^i \cdot H \pmod p$. Note that v_i is obtained from H by applying the cyclic left shift operator i times. The problem $\mathcal{P}$ is equivalent to finding a subset $S \subset [n]$ such that $|S| = h$ and $\mathsf{Ham}(\sum_{j \in S} v_j) = h$.

The search problem can be solved by enumerating all possible S in time $\binom{n}{h}\,\mathrm{poly}(n)$. A meet in the middle attack reduces the complexity of the search problem to $\binom{n}{h/2}\,\mathrm{poly}(n)$. We describe a meet in the middle attack for the cyclic subset sum problem where the goal is to find $S \subset [n], |S| = h$ such that $\sum_{j \in S} v_j = 0$. The algorithm enumerates all subsets $S' \subset [n]$ of size $h/2$, computes the list of partial sums $f(S') = \sum_{j \in S'} v_j$ and sorts the list. A solution exists if there are two subsets S', T' such that $f(S') + f(T') = 0$. The algorithm then enumerates T' and performs binary search on the sorted list to locate S' if it exists.

The above meet in the middle attack fails for the problem $\mathcal{P}$ as one needs to search for S' such that $f(S') + f(T')$ has low hamming weight. There are $\binom{n}{h/2}$ possible values of $f(S')$ to search for, thus the running time for the meet in the middle attack described above is $\binom{n}{h}$.

Guess and Win. An obvious approach to break the scheme is to try and guess one of F or G given H, and thereby obtain the private key. Notice that both H and the Hamming weight of F and G is unaffected if we multiply both F and G by a fixed power of 2. Thus, without loss of generality, we can assume that the first bit of F is 1 and then guess the remaining bits. Thus, the success probability of being able to guess one of F or G is $\frac{1}{\binom{n-1}{h-1}}$.

We could also potentially try to guess the randomness used by the encryption in order to break the scheme. Given $C = (-1)^b(AH + B) \pmod p$ for a uniformly random bit b, one can try to guess B as an h-bit string, and then if $\frac{C-B}{H} \pmod p$ has Hamming weight h, then we output $b' = 0$, if $\frac{-C-B}{H} \pmod p$ has Hamming weight less than H, then output $b' = 1$, and otherwise output a uniformly random bit b'. Note that $b' = b$ with probability 1 if B is guessed correctly, and the

probability that $b' = b$ is almost $\frac{1}{2}$ if B is guessed incorrectly. Thus, the success probability is $\frac{1}{2} + \frac{1}{\binom{n}{h}}$. A similar attempt can be made by trying to guess A and then checking the Hamming weight of $C - AH$ and $-C - AH$.

Quantum Speedup via Grover's Algorithm. If one were to attempt to break this scheme using a quantum algorithm, one could use Grover's algorithm [Gro96] to obtain a quadratic speedup over the above mentioned attacks by guessing one of F, G, A, B. Assuming that this is the best possible quantum attack on our scheme, and based on the choice of the parameters in our encryption scheme, our scheme is secure with security parameter λ.

Attacking the system if n is not a prime. We mention here that it is quite important to choose n to be a prime for our cryptosystem. There is at least a partial attack when n is not prime. Indeed if n_0 divides n, then $q = 2^{n_0} - 1$ divides $p = 2^n - 1$, and also F, G have Hamming weight at most h modulo q. Thus, given H modulo q, one can try to guess the secret key G modulo q, which can be done in $\sqrt{\binom{n_0-1}{h-1}}$ time using a quantum algorithm. This also reveals F modulo q and we can likely use it to guess F, G modulo p much faster than the brute force attack described above.

5.2 Active attacks

Active attacks and/or decryption errors attacks are powerful tools that can be used to attack our bit-by-bit encryption. We recall that the basic idea of such attacks is to ask for the decryption of incorrectly formed ciphertext and use the answers to recover information about the key.

For example, incorrect ciphertexts can be obtained by picking a random bitstring, by modifying a valid one or encrypting in a non conformant way. Here, we review the attack in the context of a single bit, but it is important to note that the encryption of many bits remain vulnerable to such attacks, even if plaintext redundancy in the style of OAEP paddings [Sho02] is added. We show in Section 6 how to withstand such attacks using appropriate checks of ciphertext validity.

For simplicity, assume that we have access to a decryption oracle. Forming pseudo ciphertexts of the form $A^*H + B^*$ with A^* and B^* with low but not conformant Hamming weights can leak information about the private key. In particular, one might incrementally add '1' bits into B^* (or A^*) until decryption transitions from 0 to $\perp$. We did not concretely write down a full working attack along this line, but it is clear that the bit-by-bit scheme would be vulnerable to such attacks.

6 Mersenne authenticated encryption plus key exchange

Since we have seen that the bit-by-bit system cannot offer resistance to chosen-ciphertext attack, we need to integrate it into a more complex scheme with this ability. A first approach would be to use an existing generic transformation for this purpose. However, this is not a simple matter, indeed, systems such as OAEP or REACT [OP01] perform checks at the plaintext level and thus cannot protect against the attack strategy of Section 5.2. The Naor-Yung paradigm [NY90, CHK10] would be more suitable but the introduction of dual-encryption and non-interactive proofs is too costly for our purpose.

In this section, we specify a full cryptosystem that achieves this level of resistance using a transformation specifically designed for our bit-by-bit encryption. This cryptosystem simultaneously encrypts a message M and performs a key exchange of a random key. Note that there is no major obstacle to turn this method into a generic transformation, this will be presented in the full paper.

In addition to the bit-by-bit Mersenne system, our transformation uses a random oracle $\mathcal{H}$. More precisely, this random oracle $\mathcal{H}$ receives as input an arbitrary bitstring and outputs a λ-bit value. As usual, every output is randomly selected whenever a fresh query is asked.

Since our transformation requires the ability to derandomize the bit-by-bit encryption process, it also needs a sampling subroutine to provide the auxiliary values used during encryption. For this purpose, we need a routine $\mathcal{S}$ that produces n-bit strings of Hamming weight h from a binary string of λ bits. This routine should be such that $\mathcal{S}(x)$ is statistically undistinguishable from a random n-bit string of weight h when x is uniformly random.

Key Generation. The key generation is identical to the bit-by-bit system and produces $\mathsf{pk} := H = \frac{F}{G} \pmod p$, and $\mathsf{sk} := G$ as before.

Encryption with key exchange. Given a message M, the algorithm proceeds as follows:

1. Pick a uniformly random λ-bit string K.
2. Encrypt bit-by-bit the concatenation $T = K\|M$. To encrypt bit number i of the string T, use the strings $A_i = \mathcal{S}(\mathcal{H}(0\|K\|[2i]_2))$ and $B_i = \mathcal{S}(\mathcal{H}(0\|K\|[2i+1]_2))$. Here, $[j]_2$ denotes the binary string representing integer j.

   Let C denote the concatenation of all encrypted values, i.e. of the sequence of numbers $(-1)^{T_i}(A_i\, H + B_i) \pmod p$.
3. Output $(C, \mathcal{H}(1\|K\|C))$ as ciphertext and K as exchanged key.

Decryption and key extraction. Given a ciphertext (C^*, m), the algorithm proceeds as follows:

- Decrypt C^* bit-by-bit and parse the decryption as $K\|M$.
- Check that m is equal to $\mathcal{H}(1\|K\|C^*)$
- Run step 2 of encryption producing C
- Check that $C = C^*$
- If both checks were successful, output (M, K). Otherwise, output $\perp$.

Note. This algorithm can also be used with the empty message in order to provide stand-alone key exchange.

Theorem 2. *Assuming that $\mathcal{H}$ is a random oracle and that the bit-by-bit scheme is semantically secure then the authenticated encryption plus key exchange is secure against adaptive chosen-ciphertext attacks.*

Sketch of proof. We need to show that chosen-ciphertext queries are not helping the adversary, i.e. that they can be simulated without significantly degrading the adversary's advantage. Once this is done, the semantic security suffices to conclude.

For this, lets consider the behavior of the decryption oracle when receiving a ciphertext (C^*, m). We want to conclude, that unless the ciphertext was produced by a procedure functionaly equivalent to the encryption specification, the decryption oracle outputs $\perp$ with overwhelming probability.

First, remark that $\mathcal{H}(1\|K\|C)$ is simply an unforgeable MAC of the string C under key K. The fact that m is correct implies that (with overwhelming probability) that the player that produced (C^*, m) made the corresponding query to $\mathcal{H}$ and thus had knowledge of the key K corresponding to the bit-by-bit decryption of the λ first encrypted bit.

The second check verifies that the decryption box does not provide anything which does not match a correct encryption following from this key K. In addition, assuming that the encryption is correctly formed, the decryption box does not return anything which cannot already be recovered from K. Indeed, when K is known, decryption can be performed without the private key G. More precisely, to decrypt bit i of a correctly formed ciphertext (given K) it suffices to regenerate the values of A_i and B_i and compare the corresponding encrypted bit with $A_iH + B_i$ and $-(A_iH + B_i)$. □

6.1 Concrete choice of parameters

Assuming that the attacks mentioned in Section 5.1 are the best possible, for getting λ-bit classical security, and $\frac{\lambda}{2}$-bit quantum security, we need to set $\binom{n-1}{h-1} > 2^\lambda$. Also, for correctness, we require $4h^2 < n$. Given our current computational capabilities, $\lambda = 128$-bit security is what we usually desire [NIS17]. Thus, choosing $h = \lfloor \sqrt{n}/2 \rfloor$, and $\lambda = \lfloor \log_2 \binom{n-1}{h-1} \rfloor$, we get the following results depending on the desired security and efficiency.

n	h	λ
1279	17	120
2203	23	174
3217	28	221
4253	32	260
9689	49	432

We mention here that even though our scheme has been defined for p being a Mersenne prime, we are not aware of any attacks even if $p = 2^n - 1$ for any large enough prime n. If we are willing to relax this requirement, then we can choose other values of n to get the desired security.

7 Conclusions and Future Work.

In this paper, we propose a simple new public-key encryption scheme. As with other public-key cryptosystems, the security of our cryptosystem relies on unproven assumptions mentioned in Definition 3. In Section 5.1, we mentioned some unsuccessful attempts we made at trying to break this scheme, and this led us to conjecture the security guarantee of our scheme. Of course, there might be other ways to attack this scheme and we urge the readers to try and find attacks on our scheme that run in time faster than $\binom{n-1}{h-1}$, or a quantum attack that runs in time faster than $\sqrt{\binom{n-1}{h-1}}$.

Acknowledgments

This research was partially funded by the Singapore Ministry of Education and the National Research Foundation, also through the Tier 3 Grant "Random numbers from quantum processes," MOE2012-T3-1-009. This work has been supported in part by the European Union's H2020 Programme under grant agreement number ERC-669891 and the French ANR Blanc program under contract ANR-12-BS02-005 (RDAM project). The second author is grateful to CQT where the work has started during his visit.

References

[BKLM11] Joppe W Bos, Thorsten Kleinjung, Arjen K Lenstra, and Peter L Montgomery. Efficient simd arithmetic modulo a mersenne number. In *Computer Arithmetic (ARITH), 2011 20th IEEE Symposium on*, pages 213–221. IEEE, 2011.

[CHK10] Ronald Cramer, Dennis Hofheinz, and Eike Kiltz. A twist on the naor-yung paradigm and its application to efficient cca-secure encryption from hard search problems. In *Theory of Cryptography, 7th Theory of Cryptography Conference, TCC 2010, Zurich, Switzerland, February 9-11, 2010. Proceedings*, pages 146–164, 2010.

[DH76] Whitfield Diffie and Martin Hellman. New directions in cryptography. *IEEE transactions on Information Theory*, 22(6):644–654, 1976.

[ES16] Lior Eldar and Peter W Shor. An efficient quantum algorithm for a variant of the closest lattice-vector problem. *arXiv preprint arXiv:1611.06999*, 2016.

[Gro96] Lov K Grover. A fast quantum mechanical algorithm for database search. *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 212–219, 1996.

[GWO+13] Lize Gu, Licheng Wang, Kaoru Ota, Mianxiong Dong, Zhenfu Cao, and Yixian Yang. New public key cryptosystems based on non-abelian factorization problems. *Security and Communication Networks*, 6(7):912–922, 2013.

[HPS98] Jeffrey Hoffstein, Jill Pipher, and Joseph Silverman. Ntru: A ring-based public key cryptosystem. *Algorithmic number theory*, pages 267–288, 1998.

[KLC+00] Ki Hyoung Ko, Sang Jin Lee, Jung Hee Cheon, Jae Woo Han, Ju-sung Kang, and Choonsik Park. New public-key cryptosystem using braid groups. In *Annual International Cryptology Conference*, pages 166–183. Springer, 2000.

[LPR10] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 1–23. Springer, 2010.

[LvTMW09] Wolfgang Lempken, Trung van Tran, Spyros S. Magliveras, and Wandi Wei. A public key cryptosystem based on non-abelian finite groups. *Journal of Cryptology*, 22(2):62–74, 2009.

[McE78] Robert J McEliece. A public-key cryptosystem based on algebraic coding theory. *Coding Thv*, 4244:114–116, 1978.

[NIS17] NIST. Post quantum crypto project. http://csrc.nist.gov/groups/ST/post-quantum-crypto/, 2017. Accessed: 2017-05-19.

[NS97] David Naccache and Jacques Stern. A new public key cryptosystem. In *EUROCRYPT 1997*, Lecture Notes in Computer Science 1233, pages 27–36, 1997.

[NY90] M. Naor and M. Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *Proceedings of the Twenty-second Annual ACM Symposium on Theory of Computing*, STOC '90, pages 427–437, New York, NY, USA, 1990. ACM.

[OP01] Tatsuaki Okamoto and David Pointcheval. REACT: rapid enhanced-security asymmetric cryptosystem transform. In *Topics in Cryptology - CT-RSA 2001, The Cryptographer's Track at RSA Conference 2001, San Francisco, CA, USA, April 8-12, 2001, Proceedings*, pages 159–175, 2001.

[Reg09] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM*, 56(6):Art. 34, 40, 2009.

[RSA78] Ronald L Rivest, Adi Shamir, and Leonard Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[RSN+01] Andrew Rukhin, Juan Soto, James Nechvatal, Miles Smid, and Elaine Barker. A statistical test suite for random and pseudorandom number generators for cryptographic applications. Technical report, DTIC Document, 2001.

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

[Sho02] Victor Shoup. OAEP reconsidered. *J. Cryptology*, 15(4):223–249, 2002.