The Mersenne Low Hamming Combination Search Problem can be reduced to an ILP Problem

Alessandro Budroni[1] and Andrea Tenti[1]

Department of Informatics, University of Bergen, Norway[1]
{alessandro.budroni,andrea.tenti}@uib.no

Abstract. In 2017, Aggarwal, Joux, Prakash, and Santha proposed an innovative NTRU-like public-key cryptosystem that was believed to be quantum resistant, based on Mersenne prime numbers $q = 2^N - 1$. After a successful attack designed by Beunardeau, Connolly, Géraud, and Naccache, the authors revised the protocol which was accepted for Round 1 of the Post-Quantum Cryptography Standardization Process organized by NIST. The security of this protocol is based on the assumption that a so-called Mersenne Low Hamming Combination Search Problem (MLHCombSP) is hard to solve. In this work, we present a reduction of MLHCombSP to an instance of Integer Linear Programming (ILP). This opens new research directions that are necessary to be investigated in order to assess the concrete robustness of such cryptosystem. We propose different approaches to perform such reduction. Moreover, we uncover a new family of weak keys, for whose our reduction leads to an attack consisting in solving $< N^3$ ILP problems of dimension 3.

Keywords: Post-Quantum Cryptography · Public-Key Cryptography · Integer Linear Programming · Mersenne-Based Cryptosystem.

1 Introduction

In [2], Aggarwal, Joux, Prakash, and Santha introduced a new public-key encryption scheme similar to the NTRU cryptosystem [12] that employs the properties of Mersenne numbers.

A Mersenne number is an integer $q = 2^N - 1$ so that N is prime. One can associate to each element in the ring $\mathbb{Z}_q$ the binary string representing the unique representative $0 \leq a < q$ of the class $[a] \in \mathbb{Z}_q$. The secret key is a pair of elements F and $G \in \mathbb{Z}_q$ so that their Hamming weight is $h < \sqrt{N/10}$. Let R be a random element of $\mathbb{Z}_q$; the public key is given by the pair $(R, T \equiv RF + G \mod q)$. The security assumption (and the mathematical problem that supports the robustness of this cryptosystem) is that it is hard to recover F and G, knowing only R and T. This assumption is called *Mersenne Low Hamming Combination Search Problem* (MLHCombSP).

This version is actually the second iteration of the cryptosystem, first presented in [1]. The security assumptions were based on a problem similar to

MLHCombSP called *Mersenne Low Hamming Ratio Search Problem* (MLHRatioSP). That has been attacked by Beunardeau et al. in [6]. There the attack is performed via a series of calls to an SVP-oracle. Its complexity has been estimated by de Boer et al. in [7]. They also showed that a Meet-in-the-Middle attack is possible using locality-sensitive hashing, which improves upon brute force. However, Beunardeau et al. turned out to be the most effective of the two. After the publications of these works, Aggarwal et al. revised the protocol [2] to prevent the above attacks from being effective against full-scale ciphers.

This protocol has been accepted to the Round 1 of the Post-Quantum Cryptography Standardization Process organized by NIST. However, it does not appear among the proposals for Round 2.

1.1 Our Contribution/Outline

In this work we present a non-trivial reduction to a relatively low-dimensional Integer Linear Programming (ILP) instance of the underlying mathematical problem of [2]. The resulting instance of ILP produces the right solution with probability p, that depends on the size of G. It is possible to perform a trade-off between the size of the ILP problem to solve and the success probability.

In section 2 we introduce notation and related work. Furthermore, we recap the Beunardeau et al. attack against [1] with a generalization to the MLHCombSP. Section 3 describes our reduction together with the success probability analysis. There we describe variations in the description of the ILP to be solved, that allow some flexibility for the attacker. In particular, one can perform a trade-off between the success probability and the dimension of the resulting ILP. The application of this trade-off is shown for two examples. In section 4 we describe a new family of weak keys and the probability of such a pair to appear. This family is obtained by performing two independent rotations on F and G so that, after these rotations, they become as small as possible. In this way the size of the set of the weak keys increases. For example, for $N = 1279$ and $h = 17$ (parameters used in [6]), a random key is weak in the sense of Beunardeau et al. with probability $\sim 2^{-34}$. It is possible to estimate that a random key becomes weak after rotations with probability $\sim 2^{-11}$.

2 Preliminaries

Definition 1 *Let N be a prime number and let $q = 2^N - 1$. Then q is called a Mersenne number. If q is also prime, then it is called Mersenne prime number.*

Let $\mathrm{seq}_N : \{0, ..., q-1\} \to \{0,1\}^N$ be the map which associates to each A the corresponding N-bits binary representation $\mathrm{seq}_N(A)$ with most-significant bit to the left.

Denote with $\mathbb{Z}_q$ the ring of integers modulo q. We extend the function seq_N also to elements in $\mathbb{Z}_q$. Let us consider an integer $0 \leq B < q$, seq_N maps $[B] \in \mathbb{Z}_q$ to the N-bits binary representation of B. We define the *Hamming weight* $w(F)$ of F as the Hamming weight of $\mathrm{seq}_N(F)$, i.e. the number of 1s in $\mathrm{seq}_N(F)$.

Lemma 1 *Let $k \geq 0$ be a positive integer, let A be an N-bits number, and let $q = 2^N - 1$. Then $\text{seq}_N(2^k A \mod q)$ corresponds to a rotation of $\text{seq}_N(A)$ of k positions to the left and $\text{seq}_N(2^{-k} A \mod q)$ corresponds to a rotation of k positions to the right.*

Proof. We prove it by induction on k. Write $\text{seq}_N(A) = (A_{N-1}, ..., A_1, A_0)$, where A_{N-1} is the most significant bit of A. Then we can represent A as

$$A = A_{N-1} \cdot 2^{N-1} + ... + A_1 \cdot 2 + A_0.$$

If we multiply A by 2 modulo q we obtain

$$\begin{aligned} 2 \cdot A &\equiv A_{N-1} \cdot 2^N + A_{N-2} \cdot 2^{N-1} + ... + A_1 \cdot 2^2 + A_0 \cdot 2 \mod q \\ &\equiv A_{N-2} \cdot 2^{N-1} + ... + A_1 \cdot 2^2 + A_0 \cdot 2 + A_{N-1} \mod q. \end{aligned}$$

Then $\text{seq}_N(2 \cdot A) = (A_{N-2}, ..., A_0, A_{N-1})$, i.e. the left rotation of 1 position of $\text{seq}_N(A)$.
By inductive hypothesis, $\text{seq}_N(2^k \cdot A)$ corresponds to the left rotation of k positions of $\text{seq}_N(A)$, then $\text{seq}_N(2^{k+1} \cdot A) = \text{seq}_N(2 \cdot 2^k \cdot A)$ corresponds to the left rotation of one position of $\text{seq}_N(2^k \cdot A)$, that is the left rotation of $k+1$ positions of $\text{seq}_N(A)$. The case right rotations of $\text{seq}_N(A)$ follows trivially. □

The security of the Aggarwal et al. cryptosystem [2] relies on the assumption that the following two problems are hard to solve.

Mersenne Low Hamming Ratio Search Problem Let $q = 2^N - 1$ be a Mersenne prime number, $h < N$ an integer, F and G two integers chosen at random from the set of N-bit numbers with Hamming weight h. Let $H < q$ be the non-negative integer such that

$$H \equiv \frac{F}{G} \mod q. \tag{1}$$

The *Mersenne Low Hamming Ratio Search Problem* (MLHRatioSP) is to find (F, G) knowing h and H.

Mersenne Low Hamming Combination Search Problem Let $q = 2^N - 1$ be a Mersenne prime number, $h < N$ an integer, R a random N-bit number, and F, G integers chosen at random from the set of N-bits numbers with Hamming weight h. Let $T < q$ be the non-negative integer such that

$$RF + G \equiv T \mod q. \tag{2}$$

The *Mersenne Low Hamming Combination Search Problem* (MLHCombSP) is to find (F, G) knowing h and the pair (R, T).

In [1], the authors suggest to choose N and h to be such that $\binom{N-1}{h-1} \geq 2^\lambda$ and $4h^2 < N$, for a desired λ-bit security level. After the publications of the attacks by Beunardeau et al. [6] and De Boer et al. [7], the authors revised the choice of the parameters ([2]) to be such that $h = \lambda$ and $10h^2 < N$.

2.1 Previous Attacks

Brute force attack In [1], Aggarwal et al. showed that a brute force attack to the MLHRatioSP would require $\binom{N-1}{h-1}$ trials. This attack consists in assuming that one of the two secret numbers, say F, has a 1 in the most significant bit (condition that can be obtained by a rotation of $\text{seq}_N(F)$). Then one should try, for every N-bits number with 1 as most significant bit and weight h, if the corresponding G through relation (1) has weight h. This approach does not apply to the MLHCombSP, which instead requires $\binom{N}{h}$ trials.

Meet-in-the-Middle attack De Boer et al. [7] showed that a Meet-in-the-Middle attack to MLHRatioSP is possible using locality-sensitive hashing with complexity $\tilde{O}\left(\sqrt{\binom{N-1}{h-1}}\right)$ on classical computers and $\tilde{O}\left(\sqrt[3]{\binom{N-1}{h-1}}\right)$ on quantum computers. This can be generalized to the MLHCombSP.

Weak Keys and Lattice attack Following the parameters' setting in [1], Beunardeau et al. found a weak key attack to the MLHRatioSP for the case when both F and G happen to have bits set to 1 only in their right halves, i.e. $F, G < \sqrt{2^N}$ [6]. This event happens with probability 2^{-2h}.

Following the above idea, Beunardeau et al. also presented a more general attack to the MLHRatioSP which consists in guessing a decomposition of F and G into windows of bits such that all the '1's are "close" to the right-most bit of such windows. Then F and G can be recovered through a lattice reduction algorithm such as LLL [13]. Even if Beunardeau et al. showed that this attack practically hits the security estimations in [1], they did not present any clear asymptotic analysis of its complexity. However, de Boer et al. [7], computed the complexity of this attack.

In [2], the authors stated that the above attack likely generalizes to the MLHCombSP case. Building directly on the work presented in [7], we show in the next subsection that this is true. However we refer the reader to [6] and [7] for a more detailed description.

2.2 The Beunardeau et al. attack on MLHCombSP

Since F is taken at random among the N-bits numbers with Hamming weight h, w.h.p. the '1' valued bits of $\text{seq}_N(F)$ do not appear in big clusters along the N possible positions. One then computes an interval-like partition $\mathcal{P}$ of $\{0, \ldots, N-1\}$ at random, i.e. each set of $\mathcal{P}$ is of the form $\{a, a+1, \ldots, b-1, b\}$, with $0 \leq a < b < N$. If each '1' valued bit of $\text{seq}_N(F)$ falls in the right-half of one of the sets of $\mathcal{P}$, then each one of them corresponds to a binary substring of $\text{seq}_N(F)$, corresponding in turn to a "small" number. Therefore, the array of these numbers can be seen as a representation of F.

Let $\mathcal{P} = \{P_1, .., P_k\}$ and $\mathcal{Q} = \{Q_1, ..., Q_l\}$ be two interval-like partitions of $\{0, ..., N-1\}$ and $(R, T) \in \mathbb{Z}_q^2$ be public parameters of an MLHCombSP instance. Let p_i, q_i be the smallest elements of P_i, Q_i respectively. Let us consider the following integer lattice.

$$\mathcal{L}_{\mathcal{P},\mathcal{Q},R,T} = \left\{ (x_1, ..., x_k, y_1, ..., y_l, u) \mid R \cdot \sum_{i=1}^{k} 2^{p_i} \cdot x_i + \sum_{j=1}^{l} 2^{q_i} \cdot y_i - uT \equiv 0 \mod q \right\}$$

The above defined lattice $\mathcal{L}_{\mathcal{P},\mathcal{Q},R,T}$ has determinant $\det(\mathcal{L}_{\mathcal{P},\mathcal{Q},R,T}) = q$ and dimension $d = k + l + 1$. Let $(F, G) \in \mathbb{Z}_q^2$ be such that $w(F) = w(G) = h$ and $RF + G \equiv T$ as in a MLHCombSP instance. Define the vector

$$\mathbf{s} = (f_1, ..., f_k, g_1, ..., g_l, 1) \in \mathcal{L}_{\mathcal{P},\mathcal{Q},R,T},$$

where $0 \leq f_i < 2^{|P_i|}$ and $0 \leq g_j < 2^{|Q_j|}$ are the unique natural numbers such that $\sum_{i=1}^{k} f_i \cdot 2^{p_i} = F$ and $\sum_{j=1}^{l} g_j \cdot 2^{q_j} = G$, where $|\cdot|$ denotes the cardinality operator. One wishes to find the vector $\mathbf{s}$ through some lattice reduction algorithm applied to $\mathcal{L}_{\mathcal{P},\mathcal{Q},R,T}$.

The lattice $\mathcal{L}_{\mathcal{P},\mathcal{Q},R,T}$ is very similar to the one defined in [7] for the MLHRatioSP and their success probability analysis of the attack holds for this case too. Therefore the following conclusions follow directly from the work of de Boer et al.

Given two partitions $\mathcal{P}$ and $\mathcal{Q}$ of $\{0, ..., N-1\}$ with block size at least $N/d + \Theta(\log N)$, where $d = k + l + 1$ with $k = |P|$ and $l = |Q|$. The success probability of finding the vector $\mathbf{s} \in \mathcal{L}_{\mathcal{P},\mathcal{Q},R,T}$ using a SVP-oracle is $2^{-2h+o(1)}$.

Remark 1 *The above attack is actually a simplified version of the attack of Beunardeau et al. Indeed, a more general attack can be made by considering the variation of partition sizes and the fraction of each partition block. This variant of the attack has success probability* $2^{-(2+\delta)h+o(1)}$*, for some small constant* $\delta > 0$ *[7].*

Remark 2 *In practice, instead of an SVP-oracle, the LLL algorithm [13] which has polynomial complexity is used. This decreases the overall complexity of the attack, but the success probability is decreased too [7].*

The above attack was made against the parameters setting contained in the first version of Aggarwal et al. work. However, as already mentioned, in the most recent version of their work the authors revisited the protocol in order to withstand it.

2.3 Integer Linear Programming

An *Integer Linear Programming* (ILP) problem in his *canonical form* is defined as follows. Given a matrix $A \in \mathbb{Q}^{m \times n}$ and two vectors $\mathbf{c} \in \mathbb{Q}^n$ and $\mathbf{b} \in \mathbb{Q}^m$, minimize (or maximise) the quantity

$$\mathbf{c}^T \mathbf{x}$$

subject to

$$\begin{cases} A\mathbf{x} \leq \mathbf{b}, \\ \mathbf{x} \geq 0, \\ \mathbf{x} \in \mathbb{Z}^n \end{cases}$$

An *ILP-oracle* is an oracle that solves any ILP instance.

Solving a general ILP is proved to be NP-hard [17]. Nevertheless, understanding the complexity of specific families of ILP problems is not an easy task: it can widely vary from case to case [18]. For example, when the number of variables is fixed, or when the problem can be reduced to a simple *Linear Programming* problem, it is proved that it has polynomial complexity [14, 20].

Nowadays there exists families of ILP solving algorithms, for example *Branch and Bound* [16], *Lagrange relaxation* [10], *Column Generation* [3], and the *Cutting Planes* [15], whose implementations [9, 11] are able to solve in practice relatively challenging instances.

3 ILP Reduction

Let R, T be two random elements of $\mathbb{Z}_q^*$. We define the map $\varphi : \mathbb{Z}_q \to \mathbb{Z}_q$ sending $X \mapsto -RX+T$. Any point on the graph of φ, namely $\{(X, \varphi(X))\}_{X \in \mathbb{Z}_q}$, satisfying the condition that both coordinates have Hamming weight equals to h is a solution to the MLHCombSP. We denote such condition as *the graph condition*.

We notice that φ is bijective, for it is the combination of two bijective functions (i.e. multiplication times a nonzero element of a field and sum with an element of the underlying group). This means that for any subset $\mathcal{U} \subseteq \mathbb{Z}_q$, the restriction $\varphi_{|\mathcal{U}}$ is injective. Hence, $|\operatorname{Im}(\varphi_{|\mathcal{U}})| = |\mathcal{U}|$. We assume that $\operatorname{Im}(\varphi_{|\mathcal{U}})$ is a random element of the family of subsets of cardinality $|\mathcal{U}|$ of $\mathbb{Z}_q$.

Let $\mathcal{V}$ be another subset of $\mathbb{Z}_q$. The probability that a random element of $\operatorname{Im}(\varphi_{|\mathcal{U}})$ is in $\mathcal{V}$ is given by $\frac{|\mathcal{V}|}{2^N-1}$. Hence the expected size of $\operatorname{Im}(\varphi_{|\mathcal{U}}) \cap \mathcal{V}$ is given by the mean of the Hypergeometric distribution [8] in $|\mathcal{U}|$ draws, from a population of size $2^N - 1$ that contains $|\mathcal{V}|$ objects that yield a success. That is:

$$\mathbb{E}(|\operatorname{Im}(\varphi_{|\mathcal{U}}) \cap \mathcal{V}|) = \frac{|\mathcal{U}||\mathcal{V}|}{2^N - 1}. \tag{3}$$

Let E_G be the number of '0' valued bits before the first '1' valued bit in $\operatorname{seq}_N(G)$. In this case, one can set $\mathcal{V} = \{2^{N-E_G-1}, \ldots, 2^{N-E_G} - 1\}$ and $|\mathcal{V}| = 2^{N-E_G}$. With such a bound on G, given a $\mathcal{U}$ of size $< 2^{E_G}$, with $F \in \mathcal{U}$, there is only one expected solution to the system of constraints:

$$\begin{cases} T - Rx \equiv y \mod q, \\ x \in \mathcal{U}, \\ y \in \mathcal{V}. \end{cases} \tag{4}$$

and one solution is certainly $x = F$, $y = G$.

Our attack is meant to find solid choices for $\mathcal{U}$ and $\mathcal{V}$ to use to solve (4).

Remark 3 *For every fixed instance of $x \in \{0, \dots, q-1\}$, there is exactly one $a \in \mathbb{Z}$ that satisfies $0 \leq T + aq - Rx < q$. In particular, this means that for every fixed instance of $x \in \{0, \dots, q-1\}$ there exists at most one $a \in \mathbb{Z}$ satisfying $2^h \leq T + aq - Rx \leq 2^N - 2^{N-h}$.*

It is possible to represent (4) in terms of integers:

$$\begin{cases} T + qa - Rx = y, \\ x \in \mathcal{U}, \\ y \in \mathcal{V}. \end{cases} \tag{5}$$

Here, there is an abuse of notation: we intend $\mathcal{U}$ as the intersection $\mathcal{U} \cap \{2^h - 1, \dots, 2^N - 2^{N-h}\}$ and $\mathcal{V}$ as the intersection $\mathcal{V} \cap \{2^h - 1, \dots, 2^N - 2^{N-h}\}$.

Remark 3 implies that the number of solutions of the system is smaller than or equal to $|\operatorname{Im}(\varphi_{|\mathcal{U}}) \cap \mathcal{V}|$. So the expected number of solutions to (5) is smaller than or equal to $\frac{|\mathcal{U}||\mathcal{V}|}{2^N - 1}$.

For some choices of $\mathcal{U}$ and $\mathcal{V}$, one can find solutions to (5) using an ILP-oracle.

Let $\mathcal{U} = \{\mathrm{l}_{x_3}, \mathrm{l}_{x_3} + 1, \dots, \mathrm{u}_{x_3} - 1, \mathrm{u}_{x_3}\}$ for some l_{x_3} and u_{x_3} and let $\mathcal{V} = \{\mathrm{l}_y, \mathrm{l}_y + 1, \dots, \mathrm{u}_y - 1, \mathrm{u}_y\}$ for some l_y and u_y. Assuming that (5) has a unique solution, then it is detected by the following ILP instance:

$$Tx_1 + qx_2 - Rx_3 = y, \tag{6}$$

with constraints

$$\begin{cases} x_1 = 1, \\ \mathrm{l}_{x_3} \leq x_3 \leq \mathrm{u}_{x_3}, \\ \mathrm{l}_y \leq y \leq \mathrm{u}_y \,. \end{cases} \tag{7}$$

Finding good choices on $\mathcal{U}$ and $\mathcal{V}$ (i.e. small and containing F and G with high probability) is difficult for the ILP instance (6). At the cost of increasing the dimension of the ILP problem to be solved, one can reduce the size of $\mathcal{U}$.

One such way is to fully exploit the fact that F has weight exactly h to establish the following ILP problem in the integer variables $x_1, x_2, x_3, n_1, \dots, n_N$:

$$Tx_1 + qx_2 - Rx_3 + 0n_1 + \cdots + 0n_N = y, \tag{8}$$

with constraints

$$\begin{cases} x_1 = 1, \\ x_3 = \sum_{i=1}^{N} n_i 2^{i-1}, \\ 0 \leq n_i \leq 1, \quad \text{for } i = 1, \dots, N \\ \sum_{i=1}^{n} n_i = h, \\ \mathrm{l}_y \leq y \leq \mathrm{u}_y \,. \end{cases} \tag{9}$$

Using these constraints results in having $\mathcal{U}$ of (5) of size $|\mathcal{U}| = \binom{N}{h}$. On the other hand, the dimension of the ILP to be solved moved from being 3 to being $N+3$. In subsections 3.2 and 3.3, we explore ways to perform trade-offs in order to choose in advance either the number of variables of the ILP to be solved or the size of $\mathcal{U}$.

3.1 Cyclic Shifts

Consider the multiplication in both sides of (2) by 2^k, for some $k > 0$,

$$2^k RF + 2^k G \equiv 2^k T \mod q. \tag{10}$$

Define $\tilde{R} \equiv 2^k R \mod q$, $\tilde{T} \equiv 2^k T \mod q$, $\tilde{F} \equiv 2^k F \mod q$ and $\tilde{G} \equiv 2^k G \mod q$. Note that $w(\tilde{F}) = w(\tilde{G}) = h$. Through (10) we can define two new MLHCombSP instances:

$$R\tilde{F} + \tilde{G} \equiv \tilde{T} \mod q, \tag{11}$$

where both F and G are rotated by k positions to the left, and

$$\tilde{R}F + \tilde{G} \equiv \tilde{T} \mod q, \tag{12}$$

where only G is rotated. By combining (11) and (12) we can rotate independently F and G. At the cost of N^2 rotations we can always find the cyclic shifts that minimizes both F and G. Performing the shifts greatly improves the probability that for small $\mathcal{U}$ and $\mathcal{V}$ of the form $\{2^{\mathrm{l}}, \ldots, 2^{\mathrm{l}+1}\}$, F and G solve (5). This results in a family of weak keys not considered in [6]. A complete analysis of the improvements is reported in Section 4.

3.2 Portion of F

As mentioned above, it is possible to reduce the dimension of the ILP (8) to be solved at the cost of increasing the size of $\mathcal{U}$. One of such methods consists in considering only the most significant bits of F in the constraints. Let γ be in the real interval $(0, 1]$. Let $\tilde{h} = \lceil \gamma h \rceil$ and let $\tilde{N} = \lceil \gamma N \rceil$. It is possible to solve the following ILP problem instead of (8):

$$Tx_1 + qx_2 - Rx_3 + 0n_1 + \cdots + 0n_{\tilde{N}} = y, \tag{13}$$

with constraints

$$\begin{cases} x_1 = 1, \\ |\sum_{i=1}^{\tilde{N}} n_i \cdot 2^{i-1} - x_3/2^{N-\tilde{N}}| < 1, \\ 0 \le n_i \le 1, \quad \text{for } i = 1, \ldots, \tilde{N} \\ \tilde{h} - t \le \sum_{i=1}^{n} n_i \le \tilde{h} + t, \\ \mathrm{l}_y \le y \le \mathrm{u}_y \,. \end{cases} \tag{14}$$

for some $0 \le t \le \tilde{h}$.

Proposition 1 *For fixed $n_1, \dots, n_{\tilde{N}}$ there exist exactly $2^{N-\tilde{N}}$ possible x_3 satisfying the first inequality in* (14).

Proof. Let write a general x_3 as $x_3 = F_{N-1}2^{N-1} + \cdots + F_0 2^0$. It follows that

$$x_3/2^{N-\tilde{N}} = F_{N-1}2^{\tilde{N}-1} + \cdots + F_{N-\tilde{N}}2^0 + F_{N-\tilde{N}-1}2^{-1} + \cdots + F_0 2^{\tilde{N}-N}.$$

We notice that $F_{N-1}, \dots, F_{N-\tilde{N}}$ are set to be equal to $n_{\tilde{N}}, \dots, n_1$ by (14), while the remaining coefficients can assume values in $\{0, 1\}$. There are exactly $2^{N-\tilde{N}}$ such x_3. □

Let us compute the size of $\mathcal{U}$ that arises from the given constraints. Thanks to Proposition 1, the size of $\mathcal{U}$ is determined only by the constraints on $n_1, \dots, n_{\tilde{N}}$. The conditions to be satisfied are:

$$\begin{cases} 0 \le n_i \le 1, \text{ for } 1 \le i \le \tilde{N}, \\ \tilde{h} - t \le \sum_{i=1}^{\tilde{N}} n_i \le \tilde{h} + t, \end{cases}$$

for some $0 \le t \le \tilde{h}$. In this scenario, the solution to the MLHCombSP is not guaranteed to be a solution to the above system. Indeed, F satisfies the above constraints if and only if its most $\tilde{N}$ significant bits contain between $\tilde{h} - t$ and $\tilde{h} + t$ '1' valued bits. This probability is given by:

$$\mathbb{P}(F \in \mathcal{U}) = \frac{\sum_{i=\tilde{h}-t}^{\tilde{h}+t} \binom{h}{i}\binom{N-h}{\tilde{N}-i}}{\binom{N}{\tilde{N}}}. \tag{15}$$

Such an $\mathcal{U}$ has size

$$|\mathcal{U}| = \sum_{i=\tilde{h}-t}^{\tilde{h}+t} \binom{\tilde{N}}{i} 2^{N-\tilde{N}}.$$

3.3 Merging

A possible approach to reduce the dimension of the ILP (8) is to merge more than one bit in a single n_i. Say, for example, that we merge the bits in pairs; this means that each one of the n_i can assume values in $\{0, 1, 2, 3\}$ and that the total weight varies between h and $2h$, as we prove in Proposition 2.

Example 1 *Let us consider $F = (00010011)$. By merging bits in pairs and assuming the MILP gives the correct solution, one gets $n_1 = (00)$, $n_2 = (01)$, $n_3 = (00)$, $n_4 = (11)$. The total sum results in $n_1 + n_2 + n_3 + n_4 = 4 \le 2h = 6$.*

Using this method, it is possible to merge an arbitrary number of bits together. Let $S = \lceil N/s \rceil$. The instance of ILP that emerges after merging bits in groups of s is the following:

$$Tx_1 + qx_2 - Rx_3 + 0n_1 + \cdots + 0n_S = y \tag{16}$$

under the conditions

$$\begin{cases} x_1 = 1, \\ \mathrm{l}_{x_2} \leq x_2 \leq \mathrm{u}_{x_2}, \\ 2^h - 1 < y < 2^N - 2^{N-h}, \\ 0 \leq n_i \leq 2^s - 1, \text{ for } 0 \leq i \leq S, \\ h \leq \sum_{i=1}^{S} n_i < 2^{s-1}h, \\ x_3 = \sum_{i=1}^{S} 2^{s(i-1)} n_i. \end{cases} \tag{17}$$

Hence the size of the ILP can be established a priori. The more bits one merges, the harder it is that the ILP will return the correct solution, for it is expected that the system of inequalities has more than one solution.

The following proposition shows that a solution $(X, \varphi(X))$ satisfying the graph condition is also a solution to the system of inequalities (17) and, therefore, it can be obtained via the ILP-oracle with the instance (16).

Proposition 2 *Let $F, G \in \mathbb{Z}_q$ so that $\varphi(F) = G$ and so that the Hamming weight of $\mathrm{seq}_N(F)$ is h. Then there exists an instance $(y, x_2, x_3, n_1, \ldots, n_S)$ with $x_3 = F$ and $y = G$ that solves the system:*

$$\begin{cases} T + x_2 q - R x_3 = y, \\ 2^h - 1 < y < 2^N - 2^{N-h}, \\ x_3 = \sum_{i=1}^{S} 2^{s(i-1)} n_i, \\ 0 \leq n_i \leq 2^s - 1, \ \textit{for } 0 \leq i \leq S, \\ h \leq \sum_{i=1}^{S} n_i \leq 2^{s-1}h. \end{cases} \tag{18}$$

Proof. The first equation and the first inequality are satisfied by the definition of φ. The second equation and the second inequality represent the fact that we are writing x_3 in base 2^s. Hence the only remaining thing to prove is that the last inequality holds.

Let $F = F(0)2^0 + \ldots + F(N-1)2^{N-1}$. We notice that $n_i = \sum_{j=0}^{s-1} F((i-1)s + j)2^j$. For the fact that $\sum_{i=0}^{N-1} F(i) = h$, we conclude that

$$\sum_{i=1}^{S} n_i = \sum_{i=1}^{S} \sum_{j=0}^{s-1} F((i-1)s + j)2^j \geq \sum_{i=1}^{S} \sum_{j=0}^{s-1} F((i-1)s + j) = h.$$

We prove the second inequality by induction on h. For $h = 1$, n_i is a string of weight 1 of s bits. That is at most 2^{s-1}.

Assuming that the inequality holds for $h - 1$. If $n_i \leq 2^{s-1}$ for every i, the inequality is satisfied. Hence we assume that there exists one j for which $n_j > 2^{s-1}$. This means that the Hamming weight of $\mathrm{seq}_s(n_j) \geq 2$. Then one gets:

$$\sum_i n_i \leq 2^s + \sum_{i \neq j} n_i.$$

The sum of the Hamming weights of $\mathrm{seq}_s(n_j)$, $j \neq i$ is at most $h-2$. By inductive hypothesis, it follows that

$$\sum_i n_i \leq 2^s + 2^{s-1}(h-2) = 2^{s-1}h.$$

The Proposition is proved. □

The following Proposition determines the size of $\mathcal{U}$ that one obtains from considering the ILP (17).

Proposition 3 *Let $\mathcal{U}$ be the set containing all $0 \leq F < q$, whose 2^s-ary representation satisfies $0 \leq n_i \leq 2^s - 1$, for $0 \leq i \leq S$ and $h \leq \sum_{i=1}^{S} n_i \leq 2^{s-1}h$. Then*

$$|\mathcal{U}| = \sum_{d=h}^{2^{s-1}h} l_{2^s}(S, d),$$

where $l_t(n, d)$ is the number of integer solutions to $z_1 + \ldots + z_n = d$, $0 \leq x_i < t$.

Proof. Let d be one of the values of $\sum_{i=1}^{S} n_i$. For each d, we consider all the possible configurations of $n_1, \ldots, n_S$. Since each of these is bounded by $2^s - 1$, the number of legitimate configurations is $l_{2^s}(S, d)$. □

Examples

In table 1 and table 2 we present the size of the resulting ILP instances depending on the value of s and the corresponding success probability in two concrete cases. We selected different choices of s and set $\mathcal{V} = \{2^{N-t-1} + 2^{h-1}, \ldots, 2^{N-t} - 2^{N-t-h}\}$ for t satisfying $\log_2(|\mathcal{U}|) + t \geq N$. The probability of $G \in \mathcal{V}$ is reported and corresponds to the success probability. Indeed, if $G \in \mathcal{V}$ then (F, G) is a solution to the system of inequalities given by the intersection of (16) with (17) and we expect it is its unique solution.

Following the attack here presented, we computed the probability that, given a fixed s, E_G is so that $\log_2(|\mathcal{U}|) + E_G \geq N$. The random variable E_G is distributed according to the negative hypergeometric distribution [4], where we are looking for the probability that the first success (first '1' valued bit) happens at the E_G-th trial, given a random sample without replacement from a population of size N containing h successes.

The parameters chosen are $N = 1279$ and $h = 17$.

Table 1.

s	Probability of success	Number of variables in ILP
1	$2^{-2.56}$	1282
2	$2^{-3.97}$	643
3	$2^{-6.13}$	430
4	$2^{-9.13}$	323
5	$2^{-12.94}$	259
6	$2^{-17.33}$	217
7	$2^{-21.73}$	186
8	$2^{-26.07}$	163
9	$2^{-30.47}$	146
10	$2^{-34.06}$	131

We notice that for these parameters, $N < 10h^2$, so it violates the guidelines given in [2]. The reason for which these where chosen is to compare the success probability with the attack by Beunardeau et al. [6], which was performed against the previous version of the protocol.

The same experiments were reproduced with $N = 1279$ and $h = 11$.

Table 2.

s	Probability of success	Number of variables in ILP
1	$2^{-1.36}$	1282
2	$2^{-1.78}$	643
3	$2^{-2.80}$	430
4	$2^{-4.29}$	323
5	$2^{-6.26}$	259
6	$2^{-8.64}$	217
7	$2^{-11.18}$	186
8	$2^{-13.71}$	163
9	$2^{-16.27}$	146
10	$2^{-18.42}$	131

Remark 4 *While solving MLHRatioSP MLHCombSP for parameters $N = 1279$ and $h = 17$ is enough to break the cryptosystem described in [1], we remark that the new security parameters suggested in [2] are $h = 256$ and $N > 10h^2$.*

Remark 5 *It is possible to generalize all the presented approaches used to account for the weight of F to account also the weight of G. However this would result in an increasing of the dimension of the ILP problem. One would nonetheless significantly increase the probability of success.*

Remark 6 *The above work can be easily adjusted in order to solve the MLHRatioSP by taking $T = 0$ and eliminating the variable x_1.*

4 A new family of weak keys

In [6] a family of weak keys was introduced for the MLHRatioSP. Those were the ones for which all the '1' valued bits appeared in the right hand side of $\text{seq}_N(F)$ and $\text{seq}_N(G)$. As noted in [2], one can break keys is this family by performing a rational reconstruction [19] of the quotient H. Aggarwal et al. also claim that the family of weak keys described in [6] extends to the MLHCombSP as well. A key in this family appears with probability 2^{-2h}.

Using the rotations described in 3.1 and the ILP instance (5), we show that this family can be extended. One can notice that many keys which have a long sequence of zeros in the middle of their bit-sequence representation are not considered as weak keys in [6]. However, we show that this is a weakness that can be exploited.

As mentioned above, one can perform up to N^2 shifts in order to get F and G as small as possible, so that it is more likely that $E_F + E_G \geq N$. Let $\mathcal{E}_F$ and $\mathcal{E}_G$ be respectively the length of the largest sequences of consecutive zeros of F and G. The distribution of such values of $\mathcal{E}_A$ is more difficult to compute and require recursion. Again, the problem is modelled as an urn problem with h white balls and $N - h$ black balls, where all the balls are samples without replacement. The probability $\mathbb{P}(\mathcal{E}_A \geq k)$ can be thought as the complementary of the probability that there are no sequences of consecutive black balls of length k. The latter, we call $\bar{p}(b, w, k)$ and is recursively defined as follows:

$$\bar{p}(b,w,k) = \begin{cases} 1 & \text{if } b \leq k, \\ 0 & \text{if } b > k \text{ and } w = 0, \\ \frac{w}{w+b-k}\bar{p}(b,w-1,k) + & \\ + \sum_{i=1}^{k-1} \left(\prod_{j=0}^{i-1} \frac{b-j}{w+b-j} \right) \frac{w}{w+b-k} \bar{p}(b-i,w-1,k) & \text{otherwise.} \end{cases}$$

Remark 7 *The probability given here is actually slightly smaller than the actual probability that the best shift has $\mathcal{E}_A \geq k$, for the current formula does not consider that the sequences of consecutive zeros can run from one extreme to the other of $\text{seq}_N(A)$. As an example, $\text{seq}_{10}(A) = (0010001000)$ will give $\mathcal{E}_A = 5$, while the $\bar{p}$ distribution will consider for A that the longest sequence of zeros is 3.*

Computing this expression is challenging even for small numbers. The estimates that we used is the following. Let Ω be the family of multisets

$$\Omega = \left\{ \{0^{a_0}, \ldots, h^{a_h}\} | a_0 \geq a_i \geq 0 \text{ for } i > 0, \sum_{i=1}^{h} a_i = N - h \right\}.$$

This family represents all the possible sequences of zeroes and ones of length N and weight h after the best shift. Let $\psi : \mathcal{Z} \to \Omega$ be the function that assigns an element of weight h in $\mathbb{Z}_q$ to the corresponding multiset in Ω. Due to symmetries, there exist $A, B \in \Omega$ so that $|\psi^{-1}(A)| \neq |\psi^{-1}(B)|$, so the probability that for a random multiset $S \in \Omega$, $a_0 = k$ is different from $\bar{p}(h, N-h, k)$. Nevertheless, experiments show that the two distributions are very similar. Hence we used the former distribution, which is easier to compute, for the numerical examples.

These computations reveal a new family of weak keys: namely, if F and G are so that $\mathcal{E}_F + \mathcal{E}_G \geq N$. One can perform N^2 rotations and guess up to $N - \lceil N/h \rceil - h$ possible $\mathcal{E}_F$ to find a unique solution to the intersection of (6) and (7), where $\mathcal{U} = \{2^{N-\mathcal{E}_F-1} + 2^{h-1}, \ldots, 2^{N-\mathcal{E}_F} - 2^{\mathcal{E}_F-h+1} + 1\}$ and where $\mathcal{V} = \{2^{\mathcal{E}_G-1} + 2^{h-1}, \ldots, 2^{\mathcal{E}_G} - 2^{\mathcal{E}_G-h+1} + 1\}$. Such solution is obtained by asking the ILP-oracle to solve instances of dimension 3.

For $N = 1279$ and $h = 17$, the expected $\mathcal{E}_A$ is ≈ 256. For these parameters and using the described estimates, one gets that $\mathbb{P}(\mathcal{E}_F + \mathcal{E}_G \geq N) \approx 2^{-11}$. This improves upon Beunardeau et al. work for which approximately 1 over 2^{34} keys is weak.

5 Conclusions and Future Work

We provide a generalization of the Beunardeu et al. attack to the case of MLH-CombSP that runs with the same time complexity, as conjectured by Aggarwal et al in [2].

We also extend the family of weak keys that should be avoided when generating the private key (F, G). Those keys can be successfully attacked with $< N^3$ queries to an ILP-oracle that solves ILP instances of dimension 3.

Results in table 1 show that, using an ILP-oracle, the success probability can be significantly higher compared to the one of the Beunardeu et al. attack [6, 7]. In practice, we would need to replace the ILP-oracle with an ILP solver. Since many practical ILP algorithms do not provide the exact solution, we expect the success probability to decrease, in the sense that, even though the system of inequalities has exactly one solution, it is not detected by the ILP solver.

In general, it is not easy to determine the complexity of an ILP instance. Unlike Linear Programming, the dimension of ILP is not determinant in establishing whether an instance is feasible or not to solve [5]. Therefore the size of the ILPs emerging from our reduction is not necessarily related to their hardness.

Unfortunately, the vast majority of the ILP solvers available does not support big numbers arithmetic. This prevented us from performing noteworthy experiments since it is an essential requirement when considering parameters that are cryptographically relevant. With a dedicated implementation it would be possible to perform such experiments that would provide empirical hints about the real complexity of those ILP instances.

Anyhow, if one wanted to use the Aggarwal et al. cryptosystem, it is advisable to investigate the nature of those ILP instances, to be sure that they do not fall into any category that allows a fast solving algorithm. We remark that ILP

problems in section 3 have only one expected possible solution and large portions of the variables are bounded by relatively tight constraints.

Aknowledgments

The authors thank Igor Semeav and Qian Guo for useful suggestions in the early stages of this work, and greatly thank Phillippe Samer for insightful discussions on ILP. The authors are also grateful to anonymous reviewers for constructive comments.

References

1. Aggarwal, D., Joux, A., Prakash, A., Santha, M.: A new public-key cryptosystem via mersenne numbers. Cryptology ePrint Archive, Report 2017/481 , version:20170530.072202 (2017)
2. Aggarwal, D., Joux, A., Prakash, A., Santha, M.: A new public-key cryptosystem via mersenne numbers. In: Shacham, H., Boldyreva, A. (eds.) Advances in Cryptology – CRYPTO 2018. pp. 459–482. Springer International Publishing, Cham (2018). https://doi.org/10.1007/978-3-319-96878-0_16
3. Appelgren, L.: A column generation algorithm for a ship scheduling problem. Transportation Science **3**, 53–68 (02 1969). https://doi.org/10.1287/trsc.3.1.53
4. Berry, K.J., Mielke Jr, P.W.: The negative hypergeometric probability distribution: Sampling without replacement from a finite population. Perceptual and motor skills **86**(1), 207–210 (1998). https://doi.org/10.2466/pms.1998.86.1.207
5. Bertsimas, D., Weismantel, R.: Optimization Over Integers. Dynamic Ideas (2005)
6. Beunardeau, M., Connolly, A., Géraud, R., Naccache, D.: On the hardness of the mersenne low hamming ratio assumption. Cryptology ePrint Archive, Report 2017/522 (2017)
7. de Boer, K., Ducas, L., Jeffery, S., de Wolf, R.: Attacks on the ajps mersenne-based cryptosystem. In: Lange, T., Steinwandt, R. (eds.) Post-Quantum Cryptography. pp. 101–120. Springer International Publishing, Cham (2018). https://doi.org/10.1007/978-3-319-79063-3_5
8. Casella, G., Berger, R.L.: Statistical inference, vol. 2. Duxbury Pacific Grove, CA (2002)
9. CPLEX Optimizer, I.: Ibm ilog cplex optimization studio (2018)
10. Fisher, M.L.: The lagrangian relaxation method for solving integer programming problems. Management science **27**(1), 1–18 (1981). https://doi.org/10.1287/mnsc.27.1.1
11. Gurobi Optimization, L.: Gurobi optimizer reference manual (2018)
12. Hoffstein, J., Pipher, J., H. Silverman, J.: Ntru: A ring-based public key cryptosystem. Algorithmic Number Theory (ANTS III) (12 1998). https://doi.org/10.1007/BFb0054868
13. Lenstra, A.K., Lenstra, H.W., Lovász, L.: Factoring polynomials with rational coefficients. Mathematische Annalen **261**(4), 515–534 (Dec 1982). https://doi.org/10.1007/BF01457454
14. Lenstra Jr, H.W.: Integer programming with a fixed number of variables. Mathematics of operations research **8**(4), 538–548 (1983). https://doi.org/10.1287/moor.8.4.538

15. Marchand, H., Martin, A., Weismantel, R., Wolsey, L.: Cutting planes in integer and mixed integer programming. Discrete Appl. Math. **123**(1-3), 397–446 (Nov 2002). https://doi.org/10.1016/S0166-218X(01)00348-1
16. Morrison, D.R., Jacobson, S.H., Sauppe, J.J., Sewell, E.C.: Branch-and-bound algorithms. Discret. Optim. **19**(C), 79–102 (Feb 2016). https://doi.org/10.1016/j.disopt.2016.01.005
17. Papadimitriou, C.H.: On the complexity of integer programming. J. ACM **28**(4), 765–768 (Oct 1981). https://doi.org/10.1145/322276.322287
18. Schrijver, A.: Theory of Linear and Integer Programming. John Wiley & Sons, Inc., New York, NY, USA (1986). https://doi.org/10.1002/net.3230200608
19. Wang, P.S.: A p-adic algorithm for univariate partial fractions. In: Proceedings of the Fourth ACM Symposium on Symbolic and Algebraic Computation. pp. 212–217. SYMSAC '81, ACM, New York, NY, USA (1981). https://doi.org/10.1145/800206.806398
20. Wolsey, L.: Integer Programming. Wiley Series in Discrete Mathematics and Optimization, Wiley (1998)