New Attacks on Mersenne Number-Based Cryptosystems: Relaxing Size Constraints

Mengce Zheng[1] and Abderrahmane Nitaj[2]

[1] Zhejiang Wanli University, Ningbo, China
mengce.zheng@gmail.com, mczheng@zwu.edu.cn
[2] Université Caen Normandie, CNRS, Normandie Univ, LMNO UMR6139, F-14000 CAEN, FRANCE
abderrahmane.nitaj@unicaen.fr

Abstract. In 2017, Aggarwal, Joux, Prakash, and Santha proposed an integer analogue of the NTRU cryptosystem, known as the AJPS cryptosystem. This scheme is based on Mersenne numbers, and is believed to be resistant to quantum attacks. This paper revisits the security of AJPS by reexamining its inherent hard problems. We propose novel attacks that apply the continued fraction-based method and the lattice-based solving strategy for modular polynomial equations, rather than directly using lattice reduction algorithms. Our methods significantly relax the size constraints on the unknown private key. Specifically, we either eliminate the need to estimate upper bounds on the unknowns or extend the attack range of vulnerable weak keys, while enhancing the success probability under unbalanced parameter settings. To validate our theoretical findings, we conduct extensive numerical experiments with parameters of various sizes. The experiments confirm both the practicality and the effectiveness of our proposed attacks.

Keywords: Attack · Continued fraction · Lattice reduction · Mersenne number · Weak keys

1 Introduction

The AJPS cryptosystem, first proposed by Aggarwal, Joux, Prakash, and Santha [1], has attracted considerable attention following its formal presentation [2] at CRYPTO 2018. This novel scheme represents a departure from traditional NTRU-based cryptosystem [16] by employing integers with sparse binary representations as secret keys instead of small-coefficient polynomials. It is conjectured to inherently resist quantum attacks, thus motivating further exploration. In the context of secure communication networks, particularly within the Internet of Things (IoT) and edge computing, the computational overhead of standard lattice-based schemes can be prohibitive. Consequently, schemes based on Mersenne primes have garnered attention as lightweight alternatives for securing data transmission channels and key exchange protocols in resource-constrained networks.

A single-bit encryption mechanism was first proposed and later refined by both Aggarwal et al. [3] and Szepieniec [22] during the NIST post-quantum standardization project. These constructions derive shared noisy secrets via sparse integers modulo a Mersenne prime and integrate error-correcting codes to counteract transmission noise. Although these proposals did not progress to subsequent rounds, which may be attributable to the nascent state of the underlying hard problems and the limited scope of their cryptanalysis, there is currently no indication of vulnerability even under quantum adversaries.

The structure of the AJPS cryptosystem can be described as follows. Let $p = 2^n - 1$ denote a Mersenne number with prime n, so that the ring $\mathbb{Z}/p\mathbb{Z}$ is isomorphic to the set of n-bit strings, where the binary sequence 1^n is identified with 0^n. Under modulo p arithmetic, an elegant correspondence is established between these integers and n-bit binary representations. In key generation, two elements f and g are randomly chosen from $\mathbb{Z}/p\mathbb{Z}$ such that each has a Hamming weight of approximately $w \approx \sqrt{n}$. Moreover, g must be invertible modulo p. The public key is then computed as $h \equiv f/g \pmod{p}$, yielding an n-bit string with an unrestricted Hamming weight, while the pair (f, g) forms the private key.

AJPS comprises a bit-level encryption scheme as well as a message-level key encapsulation mechanism (KEM). In the bit-level scheme, a Mersenne prime $p = 2^n - 1$ is chosen and two integers $f, g \in \mathbb{Z}/p\mathbb{Z}$ are selected, each with a fixed Hamming weight w, subject to the condition $n > 4w^2$. The public key is defined by $\mathsf{pk} := h \equiv f/g \pmod{p}$, and the private key is given by $\mathsf{sk} := g$.

To encrypt one bit $m \in \{0, 1\}$, two additional integers a and b, also with Hamming weight w, are randomly sampled and the ciphertext is calculated as $c = (-1)^m \cdot (ah + b)$. Its decryption proceeds by evaluating the Hamming weight $d := \mathsf{Ham}(cg)$ and deciding in favor of $m = 0$ when $d \leq 2w^2$, and $m = 1$ otherwise. This method is predicated on the observation that $cg = (-1)^m \cdot (af + bg)$, which guarantees the Hamming weight does not exceed $2w^2$ when $m = 0$.

The KEM variant involves an additional error-correcting mechanism and requires stricter parameter constraints, namely $n > 10w^2$. A random integer $r \in \mathbb{Z}/p\mathbb{Z}$ is selected, and the keys are set as $\mathsf{pk} := (r, t)$ with $t \equiv rf + g \pmod{p}$, and $\mathsf{sk} := f$. Encryption of a message $m \in \{0, 1\}^w$ involves choosing random integers a, b_1, and b_2 with Hamming weight w. The encoding algorithm $\mathcal{E} : \{0, 1\}^w \to \{0, 1\}^n$, associated with a paired error-correcting code $(\mathcal{E}, \mathcal{D})$, is then used to produce the ciphertext pair $(c_1, c_2) = (ar + b_1, (at + b_2) \oplus \mathcal{E}(m))$. Decryption is accomplished by computing $\mathcal{D}(c_1 f \oplus c_2)$, where the decoding algorithm $\mathcal{D}$ is able to recover m owing to the low Hamming distance between $c_1 f$ and $at + b_2$, as inferred from the relation $c_1 f = arf + b_1 f = a(t - g) + b_1 f = (at + b_2) + b_1 f - ag - b_2$.

In this work, we assume a strictly passive adversary capable of eavesdropping on the communication channel to intercept public keys or ciphertexts. Our cryptanalysis relies solely on these publicly available transcripts (e.g., the public key h or ciphertext pairs) to recover the secret keys or break ciphertext indistinguishability. The proposed attacks do not require any active interaction with the

communicating parties, nor do they rely on chosen-plaintext or chosen-ciphertext capabilities.

Related Work. The foundational concepts of the AJPS cryptosystem have been extended to various cryptographic constructions; however its security warrants further scrutiny. Following its initial proposal [1], subsequent studies have focused on potential vulnerabilities. Beunardeau et al. [6] introduced a lattice-based attack aimed at the underlying hard problem, achieving a time complexity on the order of $O(2^{2w})$. In parallel, work presented in [7] dissected the lattice approach and proposed a meet-in-the-middle alternative based on locality-sensitive hashing, demonstrating that the lattice-based method is superior in efficiency.

The AJPS security fundamentally relies on two computationally intractable problems. The first one is the *Mersenne Low Hamming Ratio Search Problem (MLHRSP)*, and it is formulated as follows. Let $p = 2^n - 1$ be an n-bit Mersenne prime and let w be a positive integer. Given two n-bit random strings f and g with Hamming weight w, recover f and g from the public value $h \equiv f/g \pmod{p}$. The second one is the *Mersenne Low Hamming Combination Search Problem (MLHCSP)*, and it is stated as follows. Let $p = 2^n - 1$ be an n-bit Mersenne prime, and let w be a positive integer. For a uniformly random n-bit string r, and given two random n-bit strings f and g with Hamming weight w, recover f and g from the public pair (r, t) where $t \equiv rf + g \pmod{p}$.

Beyond the aforementioned lattice-based and meet-in-the-middle attacks, additional strategies have been proposed [9,24,8]. We concentrate on lattice-based cryptanalysis in this work. We briefly review previous attacks. A core observation is exploited that if both f and g are smaller than $\sqrt{p}$, then the relation $h \equiv f/g \pmod{p}$ can be leveraged via two-dimensional lattice reduction. One can construct the lattice Λ that possesses a lattice determinant of p. The Gaussian heuristic predicts the existence of a short vector of norm roughly $\sqrt{p}$, indicating that (g, f) is a suitably short vector within this constructed lattice. Under the conditions $f, g < \sqrt{p}$, the recovery probability of the private key approximates 2^{-2w}. Extensions via random partitioning in higher-dimensional lattices have been proposed to recover the key with time complexity $O(2^{2w})$.

More recently, unbalanced parameter scenarios were addressed in [26]. By transforming the key equation $h \equiv f/g \pmod{p}$ into the bivariate form $x_1 - hx_2 \equiv 0 \pmod{p}$, Zheng and Yan exploited lattice reduction to relax the size constraints on f and g, thereby enhancing the attack's effectiveness by a factor of approximately $\sqrt{\pi} w^{3/2}/2$. Additionally, derived from the equation $c = (-1)^m \cdot (ah + b)$, an attack on the bit-level encryption utilizes a three-dimensional lattice reduction technique to recover a, b, and the plaintext m, provided that both a and b are small relative to $\sqrt{p}$. A similar approach is applicable to *MLHCSP* using the equation $t \equiv rf + g \pmod{p}$, with the same recovery probability of 2^{-2w} over the public key space.

Coron and Gini [11] proposed another attack targeting the KEM variant. Instead of extracting the private key, their approach differentiates ciphertexts between $m = 0$ or not. This attack exploits the fact that when $m = 0$ and

$\mathcal{E}(m) = 0$, resulting in ciphertext components $c_1 = ar + b_1$ and $c_2 = at + b_2$. They designed a lattice Λ that has vectors of norm around $p^{2/3}$ due to the Gaussian heuristic. Assuming that a, b_1, and b_2 are all bounded by $p^{2/3}$, lattice reduction can successfully retrieve these values, achieving a success probability on the order of $(2/3)^{3w} \approx 2^{-1.75w}$. However, existing attacks impose rather strict size constraints on the unknowns. In contrast, we aim to relax these constraints through novel attack strategies.

Our Contribution. Although the AJPS cryptosystem has been subject to various cryptanalytic efforts since its formal presentation in 2018, previous attacks impose highly restrictive size constraints on the secret unknowns. In this work, we significantly advance the state of the art by introducing new polynomial-time attacks that meaningfully relax these constraints, demonstrating that the weak key space is larger than previously estimated. Our key contributions are:

- **Heuristic-Free Continued Fraction-Based Attacks.** We introduce two continued fraction-based attack strategies for *MLHRSP* and *MLHCSP* that eliminate the need to assume explicit upper bounds on secret key sizes. Unlike previous lattice-based heuristics, these attacks rely solely on rational approximation and convergents of $\frac{h}{p}$ or $\frac{r}{p}$, yielding true polynomial-time recovery under condition $fg < p$.
- **Improved Lattice-Based Solving Strategies.** We revisit lattice-based cryptanalysis by modelling the recovery problem as solving modular polynomial equations. This allows us to handle them via a tailored lattice construction and root extraction. Our approach accommodates unbalanced secret parameters and broadens the attack range beyond classical bounds $f, g < \sqrt{p}$. While our continued fraction-based attacks are strictly heuristic-free, we note that our lattice-based methods remain inherently heuristic. This is due to the standard assumption in Coppersmith's method that the LLL-reduced polynomials are algebraically independent, a necessary condition for the subsequent root extraction phase.
- **Theoretical Analysis and Experimental Validation.** For each attack, we provide precise success probability estimates and theoretical time complexity analysis. We compare these against state-of-the-art cryptanalytic methods in both asymptotic and practical regimes. In addition, we conduct experiments not only for small w but also for larger relevant Hamming weights, demonstrating the efficiency and broader applicability.

The comparison of our proposed attacks with existing ones is given in Table 1. Furthermore, we support the theoretical propositions with extensive numerical experiments, which validate the correctness and efficacy of our proposed attacks under parameter regimes where existing ones have been shown to fail.

Organization. The remainder of this paper is organized as follows. Section 2 reviews the continued fraction-based method and the lattice-based solving strategy.

Table 1. Comparison of our proposed attacks with existing ones. The 'Constraint'-column denotes the required mathematical upper bounds (i.e., size limits) on the secret parameters such as f, g, a, b_1, b_2 for the respective attack to successfully recover the keys or plaintexts.

	Type 1: *MLHRSP*		Type 2: *MLHCSP* (bit)		Type 3: *MLHCSP* (KEM)	
	[6,7,11]	[26], Ours	[11]	Ours	[11]	Ours
Probability	2^{-2w}	$\sqrt{\pi}w^{3/2}2^{-2w-1}$	2^{-2w}	$w^{3/2}2^{-2w}$	$2^{-1.75w}$	$w^3 2^{-1.75w}$
Constraint	$f, g < \sqrt{p}$	$fg < p$	$f, g < \sqrt{p}$	$fg < p$	$a, b_1, b_2 < p^{2/3}$	$ab_1b_2 < p^2$
Time	$\mathrm{poly}(\log p)$	$\mathrm{poly}(\log p)$	$\mathrm{poly}(\log p)$	$\mathrm{poly}(\log p)$	$\mathrm{poly}(\log p)$	$\mathrm{poly}(\log p)$

In Section 3 and Section 4, we present new attacks on *MLHRSP* and *MLHCSP* along with their success probability analysis. Section 5 reports the outcomes of our numerical experiments. Finally, Section 6 concludes the paper.

2 Preliminaries

To establish the foundation for our new attack strategies, we recall the continued fraction-based method and the lattice-based solving strategy.

2.1 Continued Fractions

Let γ be a nonzero real number. For $i \geq 0$, define the series x_i and a_i as follows

$$x_0 = \gamma, \quad a_i = \lfloor x_i \rfloor, \quad \text{if } \; x_i - a_i \neq 0, \quad x_{i+1} = \frac{1}{x_i - a_i},$$

where $\lfloor x_i \rfloor$ is the integer part of x_i. Then $a_0 \in \mathbb{Z}$ and $a_1, a_2, \ldots$ are positive integers, and are called the partial quotients. They allow us to express γ as a continued fraction expansion given by

$$\gamma = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{\ddots}}},$$

which is commonly abbreviated as $\gamma = [a_0, a_1, a_2, \ldots]$. The i-th convergent of γ is the rational number

$$\frac{p_i}{q_i} = [a_0, a_1, \ldots, a_i].$$

The convergents can be computed recursively by the Euler-Wallis theorem, specifically

$$p_0 = a_0, \quad p_1 = a_0 a_1 + 1, \quad p_i = a_i p_{i-1} + p_{i-2}, \quad i \geq 2,$$

$$q_0 = 1, \quad q_1 = a_1, \quad q_i = a_i q_{i-1} + q_{i-2}, \quad i \geq 2.$$

When $\gamma = \frac{r}{p}$ is rational, its continued fraction expansion is finite and its convergents are polynomial-time computable. Moreover, the following properties are satisfied.

1. The convergents are ordered as
$$\frac{p_0}{q_0} < \frac{p_2}{q_2} < \cdots < \frac{r}{p} < \cdots < \frac{p_3}{q_3} < \frac{p_1}{q_1}. \tag{1}$$
2. For $i \geq 0$, define
$$r_i = (-1)^i (q_i r - p_i p).$$
Then the following facts hold:
$$\begin{aligned} &r_i > r_{i+1} \geq 0, \quad p_i < p_{i+1}, \quad q_i < q_{i+1}, \\ &r_i q_{i+1} + r_{i+1} q_i = p, \quad r_i p_{i+1} + r_{i+1} p_i = r, \\ &p_{i+1} q_i - p_i q_{i+1} = (-1)^i, \quad r_{i+1} = r_{i-1} - a_{i+1} r_i. \end{aligned} \tag{2}$$

A fundamental theorem, due to Legendre [18], certifies that a rational number $\frac{x}{y}$ is amongst the convergents of γ when $\left|\gamma - \frac{x}{y}\right| < \frac{1}{2y^2}$. An extension of this result was first stated by Fatou, and proven by Grace in the following form.

Theorem 1 ([12,14]). *Let $\gamma \in \mathbb{R}$ and let $x, y \in \mathbb{Z}$ with $\gcd(x, y) = 1$. If*
$$\left|\gamma - \frac{x}{y}\right| < \frac{1}{y^2},$$
then there exist three consecutive convergents $\frac{p_i}{q_i}$, $\frac{p_{i+1}}{q_{i+1}}$, $\frac{p_{i+2}}{q_{i+2}}$ of γ such that
$$\frac{x}{y} \in \left\{ \frac{p_i}{q_i}, \frac{p_{i+1} + p_i}{q_{i+1} + q_i}, \frac{p_{i+2} - p_{i+1}}{q_{i+2} - q_{i+1}} \right\}.$$

2.2 Lattice-Based Solving Strategy

We introduce lattices, the celebrated LLL lattice reduction algorithm [19], and Coppersmith's lattice-based method [10] (further refined by Howgrave-Graham [17]). For a more detailed discussion, readers are referred to [20,21,25].

A *lattice* Λ is defined as the set of all integer linear combinations of a set of linearly independent vectors $\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_\omega$ in $\mathbb{R}^n$, that is $\Lambda = \{\sum_{i=1}^{\omega} z_i \boldsymbol{b}_i : z_i \in \mathbb{Z}\}$. The *determinant* of the lattice, denoted by $\det(\Lambda)$, is given by $\sqrt{\det(BB^{\mathsf{T}})}$, where the rows of the matrix B are the basis vectors $\boldsymbol{b}_i$ and B^{T} is its transpose. In the case of a full-rank lattice (i.e., $\omega = n$), this simplifies to $\det(\Lambda) = |\det(B)|$.

The LLL algorithm [19] is a core mathematical tool for approximating short vectors in a lattice. Its efficiency in reducing a given lattice basis is captured in the following lemma.

Lemma 1 ([19,20]). *For any ω-dimensional lattice Λ generated by $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_\omega$, the LLL algorithm returns a reduced basis $(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_\omega)$ satisfying*

$$\|\boldsymbol{v}_i\| \leq 2^{\frac{\omega(\omega-1)}{4(\omega+1-i)}} \det(\Lambda)^{\frac{1}{\omega+1-i}}, \quad \textit{for } i = 1, 2, \ldots, \omega.$$

The algorithm runs in time polynomial in ω and the logarithm of the maximum absolute value of the components of the basis vectors.

Howgrave-Graham's reformulation [17] provides a criterion to certify that a solution modulo some integer is indeed a true solution over the integers. Consider an integer polynomial $h(x_1, \ldots, x_n) = \sum c_{i_1,\ldots,i_n} x_1^{i_1} \cdots x_n^{i_n}$ with the Euclidean norm defined as $\|h\| := \sqrt{\sum |c_{i_1,\ldots,i_n}|^2}$. The following lemma encapsulates the key idea:

Lemma 2 ([17]). *Let $h(x_1, \ldots, x_n) \in \mathbb{Z}[x_1, \ldots, x_n]$ be a polynomial with at most ω monomials. Given positive integers $R, X_1, \ldots, X_n$, if the two conditions $h(x_1^\star, \ldots, x_n^\star) \equiv 0 \pmod{R}$ with $|x_i^\star| \leq X_i$ and $\|h(X_1 x_1, \ldots, X_n x_n)\| < \frac{R}{\sqrt{\omega}}$ hold, then $h(x_1^\star, \ldots, x_n^\star) = 0$ holds over the integers.*

The combination of the LLL algorithm and Howgrave-Graham's lemma offers a powerful framework for solving modular polynomial equations. Suppose one computes the first ℓ reduced vectors $(\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_\ell)$ from the lattice Λ. By Lemma 1, for each $1 \leq i \leq \ell$ one has

$$\|\boldsymbol{v}_i\| \leq 2^{\frac{\omega(\omega-1)}{4(\omega+1-i)}} \det(\Lambda)^{\frac{1}{\omega+1-i}} \leq 2^{\frac{\omega(\omega-1)}{4(\omega+1-\ell)}} \det(\Lambda)^{\frac{1}{\omega+1-\ell}}.$$

To guarantee that one of these vectors leads to a polynomial that vanishes over the integers, it suffices to enforce $2^{\frac{\omega(\omega-1)}{4(\omega+1-\ell)}} \det(\Lambda)^{\frac{1}{\omega+1-\ell}} < \frac{R}{\sqrt{\omega}}$, which implies $\det(\Lambda) < R^{\omega+1-\ell} 2^{-\frac{\omega(\omega-1)}{4}} \omega^{-\frac{\omega+1-\ell}{2}}$. Since typically $\ell < \omega \ll R$, this condition can be asymptotically approximated by

$$\det(\Lambda) < R^\omega. \tag{3}$$

Let $h(x_1, \ldots, x_n)$ be an n-variate integer polynomial and $h(x_1^\star, \ldots, x_n^\star) \equiv 0 \pmod{u}$ with known bounds $|x_i^\star| \leq X_i$. The goal is to recover the tuple $(x_1^\star, \ldots, x_n^\star)$ efficiently. We outline the standard procedure employed in the lattice-based solving strategy as follows.

Polynomials Construction: Generate *shift polynomials* $s_k(x_1, \ldots, x_n)$ for $1 \leq k \leq \omega$ so that each satisfies $s_k(x_1^\star, \ldots, x_n^\star) \equiv 0 \pmod{u^m}$, for a suitable exponent m. A common choice is $s_k(x_1, \ldots, x_n) := x_1^{i_{k1}} \cdots x_n^{i_{kn}} h^{j_k} u^{m-j_k}$, where $i_{k1}, \ldots, i_{kn}, j_k$ are non-negative integers. Besides, R mentioned above is hence equal to u^m. **Lattice Generation:** Represent the monomial-wise coefficients of the scaled shift polynomials $s_k(X_1 x_1, \ldots, X_n x_n)$ as row vectors of a basis matrix B, thereby generating a lattice Λ. This lattice basis matrix can be full-rank and triangular under suitable term orders. **Lattice Reduction:** Apply the LLL algorithm to B to obtain a reduced basis $(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_\ell)$ (with $\ell \geq n$). Convert the reduced vectors into corresponding polynomials $v_k(x_1, \ldots, x_n)$ sharing the

common root $(x_1^\star, \ldots, x_n^\star)$ over the integers. The strategy succeeds provided that the condition (3) is satisfied. **Root Extraction:** Assuming the obtained polynomials are algebraically independent, one can compute the common root using techniques such as resultant method or Gröbner basis computation [5].

The recovery of the desired root primarily hinges on the efficiency of both the lattice reduction and the subsequent root extraction, both of which can be executed in polynomial time relative to the input size. Note that it is generally assumed that the LLL-reduced polynomials possess the required independence in practical lattice-based attacks. While there exist counterexamples in limited cases, for the sake of efficiency we adopt this assumption throughout the paper.

3 Continued Fraction-Based Attacks

3.1 The First Attack

The first attack deals with the *MLHRSP* in AJPS. This attack is achieved by rewriting the key equation $h \equiv f/g \pmod p$ as $f = hg - kp$ for an unknown integer k. This observation suggests searching for the ratio $\frac{k}{g}$ among the convergents in the continued fraction expansion of the known quantity $\frac{h}{p}$. In particular, we have the following result.

Theorem 2. *Let $p = 2^n - 1$ be an n-bit Mersenne prime and let w be a positive integer. Suppose the secret keys f and g with Hamming weight w satisfy the bounds $f \leq p^{\xi_1}$ and $g \leq p^{\xi_2}$. Given a positive integer $h < p$ satisfying $h \equiv f/g \pmod p$, f and g can be found in time polynomial in n if $fg < p$, or equivalently $\xi_1 + \xi_2 < 1$.*

Proof. Suppose that $h \equiv f/g \pmod p$. Then, there exists an unknown integer k such that $f \equiv hg - kp$. If $fg < p$, then

$$\left| \frac{h}{p} - \frac{k}{g} \right| = \frac{f}{pg} < \frac{1}{g^2}.$$

By Theorem 1, the convergents of the continued fraction expansion of $\frac{h}{p}$ yield unique rational approximations. Hence, k and g can be recovered from these convergents. More sepecifically, for one of such convergents, we have

$$(k, g) \in \{(p_i, q_i), (p_{i+1} + p_i, q_{i+1} + q_i), (p_{i+2} - p_{i+1}, q_{i+2} - q_{i+1})\}.$$

Once g and k are known, f can be computed as $f = hg - kp$. The couple (f, g) is the desired solution if their Hamming weight is w.

The attack time complexity mainly depends on the continued fraction expansion, which is a polynomial in the bit-length of inputs. Therefore, the running time is a polynomial in $\log p = n$. □

Though we reproduce the result of the previous attack on *MLHRSP* [26], our new attack strategy shows that the continued fraction-based method is more efficient since it eliminates the need to estimate upper bounds on the unknowns.

This implies that we do not require the prior knowledge of ξ_1 and ξ_2 such that $f < p^{\xi_1}$, and $g < p^{\xi_2}$. The success probability $\Pr_1$ of the above attack is approximately $\sqrt{\pi} w^{3/2} 2^{-2w-1}$, which is identical to the previous one [26]. So we skip the detailed analysis here.

3.2 The Second Attack

The second attack deals with the *MLHCSP* in AJPS. We present a new attack on the congruence $t \equiv rf + g \pmod{p}$ using the continued fraction method. The following result presents an attacking strategy to solve $t \equiv rf + g \pmod{p}$ when $fg < p$.

Theorem 3. *Let $p = 2^n - 1$ be an n-bit Mersenne prime and let w be a positive integer. Suppose the secret keys f and g with Hamming weight w satisfy the bounds $f \le p^{\xi_1}$ and $g \le p^{\xi_2}$. Given positive integers $r, t < p$ satisfying $t \equiv rf + g \pmod{p}$, f and g can be found in time polynomial in n if $fg < p$, or equivalently $\xi_1 + \xi_2 < 1$.*

Proof. Let f and g be integers satisfying $t \equiv rf + g \pmod{p}$, equivalently $t = rf + g - kp$ for some integer k. Let

$$\frac{p_0}{q_0} < \frac{p_2}{q_2} < \cdots < \frac{r}{p} < \cdots < \frac{p_3}{q_3} < \frac{p_1}{q_1}$$

be the convergents of $\frac{r}{p}$. For $i \ge 0$, we define

$$r_i = (-1)^i (q_i r - p_i p). \tag{4}$$

By the properties in (2) we have $r_i > r_{i+1} \ge 0$, $p_i < p_{i+1}$, $q_i < q_{i+1}$, and

$$r_i q_{i+1} + r_{i+1} q_i = p. \tag{5}$$

Therefore, the differences r_i in (4) mean that for $i \ge 0$, we have

$$q_i r - (-1)^i r_i = p_i p, \quad q_{i+1} r + (-1)^i r_{i+1} = p_{i+1} p.$$

For arbitrary integers a and b, we combine these equalities to obtain

$$(aq_i - bq_{i+1})r - (-1)^i (ar_i + br_{i+1}) = (ap_i - bp_{i+1})p. \tag{6}$$

Identifying the left-hand side of (6) with $rf + g - t = kp$, we then set

$$\begin{cases} aq_i - bq_{i+1} & = f, \\ ap_i - bp_{i+1} & = k, \\ -(-1)^i (ar_i + br_{i+1}) & = g - t. \end{cases}$$

Since $p_{i+1} q_i - p_i q_{i+1} = (-1)^i$ by (2), the integers a and b are uniquely determined by f and k using the first two equations, and the deduced continued fraction expansion of $\frac{r}{p}$. Using the first and the third equations, we get

$$a = \frac{(-1)^i (t-g) q_{i+1} + f r_{i+1}}{r_i q_{i+1} + r_{i+1} q_i}, \quad b = \frac{(-1)^i (t-g) q_i - f r_i}{r_i q_{i+1} + r_{i+1} q_i}.$$

Using (5), these can be rewritten as

$$a = \frac{(-1)^i t q_{i+1}}{p} + \frac{-(-1)^i g q_{i+1} + f r_{i+1}}{p},$$
$$b = \frac{(-1)^i t q_i}{p} + \frac{-(-1)^i g q_i - f r_i}{p}.$$

Assume $fg < p$ and let i be the largest index such that $q_i \le f < q_{i+1}$. Then we have

$$g < \frac{p}{f} \le \frac{p}{q_i}.$$

We can bound the deviation of a from its main term:

$$\left| a - \frac{(-1)^i t q_{i+1}}{p} \right| = \frac{\left| -(-1)^i g q_{i+1} + f r_{i+1} \right|}{p} \le \frac{g q_{i+1} + f r_{i+1}}{p}$$
$$< \frac{\frac{p q_{i+1}}{q_i} + q_{i+1} r_{i+1}}{p} = \frac{q_{i+1}}{q_i} + \frac{q_{i+1} r_{i+1}}{p}.$$

Hence, it implies that

$$a = \left\lfloor \frac{(-1)^i t q_{i+1}}{p} \right\rceil + \tilde{a}, \quad |\tilde{a}| < \frac{q_{i+1}}{q_i} + \frac{q_{i+1} r_{i+1}}{p}.$$

Moreover, note that

$$\frac{q_{i+1}}{q_i} + \frac{q_{i+1} r_{i+1}}{p} < \frac{q_{i+1}}{q_i} + \frac{q_{i+1} r_i}{p} = \frac{q_{i+1}}{q_i} + 1 - \frac{q_i r_{i+1}}{p} < \frac{q_{i+1}}{q_i} + 1.$$

A similar bound holds for b with its deviation term:

$$\left| b - \frac{(-1)^i t q_i}{p} \right| = \frac{\left| -(-1)^i g q_i - f r_i \right|}{p} \le \frac{g q_i + f r_i}{p}$$
$$< \frac{p + q_{i+1} r_i}{p} = \frac{p + p - q_i r_{i+1}}{p} = 2 - \frac{q_i r_{i+1}}{p}.$$

This implies that

$$b = \left\lfloor \frac{(-1)^i t q_i}{p} \right\rceil + \tilde{b}, \quad |\tilde{b}| \le 1.$$

Thus, a and b lie in short integer intervals around their principal terms and can be recovered by a small exhaustive search for the appropriate index i. This completes the proof. □

We remark that the GLV method [13] for solving $t \equiv rf + g \pmod{p}$ applies when f and g are balanced with $f \approx g \approx \sqrt{p}$. The above proposed approach also recovers such balanced solutions and can therefore be viewed as an extension of the GLV method. In addition, we provide the following attack algorithm for completeness.

Algorithm 1 Continued fraction-based attack algorithm on *MLHCSP* (bit)

Input: $p = 2^n - 1$, r, t, w
Output: f, g such that $fg < p$, $t \equiv rf + g \pmod p$, and $\mathsf{HW}(f) = \mathsf{HW}(g) = w$

Compute continued fraction convergents $\frac{p_i}{q_i}$ of $\frac{r}{p}$ for $i = 0, 1, \ldots, l$
for $i = 1$ **to** l **do**
  $r_i \leftarrow (-1)^i (q_i r - p_i p)$
end for
for $i = 1$ **to** $l - 1$ **do**
  $\bar{a} \leftarrow \lfloor (-1)^i t q_{i+1}/p \rceil$, $\hat{a} \leftarrow \lceil q_{i+1}/q_i + q_{i+1} r_{i+1}/p \rceil$
  $\bar{b} \leftarrow \lfloor (-1)^i t q_i/p \rceil$, $\hat{b} \leftarrow \lceil 2 - q_i r_{i+1}/p \rceil$
  for $\tilde{a} = -\hat{a}$ **to** $\hat{a}$ **do**
    $a \leftarrow \bar{a} + \tilde{a}$
    for $\tilde{b} = -\hat{b}$ **to** $\hat{b}$ **do**
      $b \leftarrow \bar{b} + \tilde{b}$
      $f \leftarrow aq_i - bq_{i+1}$
      $g \leftarrow t - (-1)^i (ar_i + br_{i+1}) \pmod p$
      if $\mathsf{HW}(f) = \mathsf{HW}(g) = w$ **and** $t \equiv rf + g \pmod p$ **then**
        return f, g ▷ solution found!
      end if
    end for
  end for
end for

Observe that there are $l = \mathcal{O}(\log p)$ convergents of $\frac{r}{p}$, and that the ratio $\frac{q_{i+1}}{q_i}$ in $\tilde{a}$ typically concentrates around the Khintchine constant ≈ 2.6854 (see [4]). Hence, one can assume $|\tilde{a}| \leq 3$, while $|\tilde{b}| \leq 1$. Therefore, the search in Algorithm 1 is over a small constant-sized neighborhood for each of the $\mathcal{O}(\log p)$ indices, and the overall complexity is

$$7 \times 3 \times (l - 1) = \mathcal{O}(\log p),$$

which is polynomial in $\log p = n$.

4 Lattice-Based Attacks

4.1 The Third Attack

Assume that we are given the public data r and $t \equiv rf + g \pmod p$, where the unknowns f and g satisfy certain size restrictions. Our strategy is to construct a bivariate polynomial

$$h(x_1, x_2) = rx_1 + x_2 - t,$$

so that the congruence

$$h(x_1, x_2) \equiv 0 \pmod p \tag{7}$$

admits the solution $(x_1^\star, x_2^\star) = (f, g)$.

For analysis, we assume that the unknowns are bounded by $X_1 = p^{\xi_1}$ and $X_2 = p^{\xi_2}$, and we set a positive integer parameter m that will govern the construction of shift polynomials. For non-negative integers i_1 and i_2 satisfying $i_1 + i_2 \leq m$, we define the shifts by

$$s_{[i_1,i_2]}(x_1, x_2) := x_1^{i_1} h^{i_2}(x_1, x_2)\, p^{m-i_2}.$$

In this attack, the parameter R involved in Lemma 2 and the discussion that follows it is equal to p^m.

The next step is to transform the coefficient vectors of the scaled polynomials $s_{[i_1,i_2]}(X_1x_1, X_2x_2)$ into row vectors of a lattice basis matrix B. For this purpose, we fix a monomial order: for monomials $x_1^{i_1}x_2^{i_2}$ and $x_1^{i_1'}x_2^{i_2'}$, we say $x_1^{i_1}x_2^{i_2} \prec x_1^{i_1'}x_2^{i_2'}$ if $i_1 + i_2 < i_1' + i_2'$, or $i_1 + i_2 = i_1' + i_2'$ with $i_1 > i_1'$. This ordering also applies to the shift polynomials $s_{[i_1,i_2]}$; their leading monomial is exactly $x_1^{i_1}x_2^{i_2}$.

The number of constructed shift polynomials, which determines the lattice dimension, is

$$\omega = \sum_{i_1=0}^{m} \sum_{i_2=0}^{m-i_1} 1 = \frac{(m+1)(m+2)}{2}.$$

An illustrative toy example with $m = 3$ produces a basis matrix B whose entries are given by the coefficients of the polynomials in the order induced by the above monomial ordering:

$$\left(\begin{array}{c|cccccccccc}
 & 1 & x_1 & x_2 & x_1^2 & x_1x_2 & x_2^2 & x_1^3 & x_1^2x_2 & x_1x_2^2 & x_2^3 \\ \hline
s_{[0,0]} & p^3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
s_{[1,0]} & 0 & X_1p^3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
s_{[0,1]} & -tp^2 & rX_1p^2 & X_2p^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
s_{[2,0]} & 0 & 0 & 0 & X_1^2p^3 & 0 & 0 & 0 & 0 & 0 & 0 \\
s_{[1,1]} & 0 & -tX_1p^2 & 0 & rX_1^2p^2 & X_1X_2p^2 & 0 & 0 & 0 & 0 & 0 \\
s_{[0,2]} & t^2p & -2rtX_1p & -2tX_2p & r^2X_1^2p & 2rX_1X_2p & X_2^2p & 0 & 0 & 0 & 0 \\
s_{[3,0]} & 0 & 0 & 0 & 0 & 0 & 0 & X_1^3p^3 & 0 & 0 & 0 \\
s_{[2,1]} & 0 & 0 & 0 & -tX_1^2p^2 & 0 & 0 & rX_1^3p^2 & X_1^2X_2p^2 & 0 & 0 \\
s_{[1,2]} & 0 & t^2X_1p & 0 & -2rtX_1^2p & -2tX_1X_2p & 0 & r^2X_1^3p & 2rX_1^2X_2p & X_1X_2^2p & 0 \\
s_{[0,3]} & -t^3 & 3rt^2X_1 & 3t^2X_2 & -3r^2tX_1^2 & -6rtX_1X_2 & -3tX_2^2 & r^3X_1^3 & 3r^2X_1^2X_2 & 3rX_1X_2^2 & X_2^3
\end{array}\right).$$

In this matrix, the diagonal entries are exactly $X_1^{i_1}X_2^{i_2}p^{m-i_2}$ for $0 \leq i_1+i_2 \leq m$.

Following the standard lattice-based solving strategy, the determinant of the lattice Λ generated by the rows of B can be expressed as

$$\det(\Lambda) = X_1^{\sigma_1}X_2^{\sigma_2}p^{\sigma_p},$$

where

$$\sigma_1 = \sum_{i_1=0}^{m} \sum_{i_2=0}^{m-i_1} i_1 = \frac{m(m+1)(m+2)}{6},$$

$$\sigma_2 = \sum_{i_1=0}^{m} \sum_{i_2=0}^{m-i_1} i_2 = \frac{m(m+1)(m+2)}{6},$$

$$\sigma_p = \sum_{i_1=0}^{m} \sum_{i_2=0}^{m-i_1} (m - i_2) = \frac{m(m+1)(m+2)}{3}.$$

The lattice-based solving strategy succeeds provided that the condition holds:

$$\det(\Lambda) < R^{\omega}, \quad \text{with } R = p^m.$$

Inserting the expressions above and substituting $X_1 = p^{\xi_1}$ and $X_2 = p^{\xi_2}$, one obtains the constraint

$$X_1^{\frac{m(m+1)(m+2)}{6}} X_2^{\frac{m(m+1)(m+2)}{6}} p^{\frac{m(m+1)(m+2)}{3}} < p^{m \cdot \frac{(m+1)(m+2)}{2}}.$$

After simplifying the exponents with respect to p, this reduces to the inequality

$$\frac{1}{6} \cdot (\xi_1 + \xi_2) + \frac{1}{3} < \frac{1}{2},$$

and further leads to

$$\xi_1 + \xi_2 < 1. \tag{8}$$

Once a sufficient number of integer polynomials $v_k(x_1, x_2)$ (each vanishing at the root (f, g)) are extracted via lattice reduction and subsequent techniques (e.g., resultant computation), the common root $(x_1^\star, x_2^\star) = (f, g)$ can be recovered efficiently. Note that the overall time complexity is mainly dominated by the LLL algorithm and is polynomial in the fixed parameter m and $\log p = n$. The overall complexity of our lattice-based attack is polynomial in n and independent of w. The former analysis is summarized in the following proposition.

Proposition 1. *Let $p = 2^n - 1$ be an n-bit Mersenne prime and let w be a positive integer. Suppose the secret keys f and g with Hamming weight w satisfy the bounds $f \le p^{\xi_1}$ and $g \le p^{\xi_2}$. Given r and t satisfying $t \equiv rf + g \pmod p$, f and g can be recovered in time polynomial in n provided that $\xi_1 + \xi_2 < 1$.*

The condition $\xi_1 + \xi_2 < 1$ implies in particular that $fg < p$. This generalizes previous results in which both f and g were assumed to be less than $\sqrt{p}$, and extends the attack to unbalanced choices of f and g.

We analyze the success probability of the above attack. Representing f and g as n-bit strings with w ones, the earlier attack by Beunardeau et al. [6] assumed that these ones lie within the lower $\lfloor n/2 \rfloor$ bit positions, leading to a success probability roughly given by

$$\frac{\binom{\lfloor n/2 \rfloor}{w}\binom{\lfloor n/2 \rfloor}{w}}{\binom{n}{w}\binom{n}{w}} = \left(\frac{\lfloor n/2 \rfloor!(n-w)!}{n!(\lfloor n/2 \rfloor - w)!}\right)^2 \approx 2^{-2w}.$$

In contrast, our new attack achieves a success probability

$$\Pr_2 = \frac{\sum_{k=w}^{n-w} \binom{k}{w}\binom{n-k}{w}}{\binom{n}{w}\binom{n}{w}}.$$

Using a combinatorial identity based on binomial coefficients from combinatorial mathematics [15],

$$\sum_{k=i}^{n-j} \binom{k}{i}\binom{n-k}{j} = \binom{n+1}{i+j+1},$$

we deduce

$$\mathrm{Pr}_2 = \frac{\binom{n+1}{2w+1}}{\binom{n}{w}^2}.$$

Using the assumption that $w \approx \sqrt{n}$ (i.e., $n \approx w^2$) as $n \to \infty$, the asymptotic improvement factor r_2 can be derived as

$$\mathsf{r}_2 = \frac{\binom{n+1}{2w+1}}{\binom{\lfloor n/2 \rfloor}{w}^2} = \frac{(n+1)!(w!(\lfloor n/2 \rfloor - w)!)^2}{(2w+1)!(n-2w)!(\lfloor n/2 \rfloor !)^2} \approx w^{3/2}.$$

More details can be found in Appendix A.1. Thus asymptotically, the success probability of this attack is approximately $\mathrm{Pr}_2 \approx w^{3/2} 2^{-2w}$.

4.2 The Fourth Attack

We consider the setting of breaking ciphertext indistinguishability in the KEM variant. In this case, the ciphertext components satisfy $c_1 = ar + b_1$ and $c_2 = at + b_2$ with the unknowns a, b_1, and b_2 subject to certain bounds. Define two bivariate polynomials by

$$h_1(x_1, x_2) = rx_1 + x_2 - c_1 \quad \text{and} \quad h_2(x_1, x_3) = tx_1 + x_3 - c_2.$$

Then the congruences

$$\begin{cases} h_1(x_1, x_2) \equiv 0 \pmod{p}, \\ h_2(x_1, x_3) \equiv 0 \pmod{p}, \end{cases} \tag{9}$$

are simultaneously satisfied by the solution $(x_1^\star, x_2^\star, x_3^\star) = (a, b_1, b_2)$.

Let the respective upper bounds on the unknowns be $X_1 = p^{\xi_1}$, $X_2 = p^{\xi_2}$, and $X_3 = p^{\xi_3}$. For a fixed m and non-negative integers i_1, i_2, i_3 with $i_1 + i_2 + i_3 \leq m$, we form the shift polynomials

$$s_{[i_1,i_2,i_3]}(x_1, x_2, x_3) := x_1^{i_1} h_1^{i_2}(x_1, x_2) h_2^{i_3}(x_1, x_3)\, p^{m-i_2-i_3}.$$

Here, $R = p^m$ as before. Then the coefficients of the scaled shift polynomials $s_{[i_1,i_2,i_3]}(X_1 x_1, X_2 x_2, X_3 x_3)$ are arranged as row vectors in a lattice basis matrix B, which generates the corresponding lattice Λ.

We adopt a monomial order on $x_1^{i_1} x_2^{i_2} x_3^{i_3} \prec x_1^{i_1'} x_2^{i_2'} x_3^{i_3'}$ defined by:

1. $i_1 + i_2 + i_3 < i_1' + i_2' + i_3'$, or
2. $i_1 + i_2 + i_3 = i_1' + i_2' + i_3'$ with $i_1 > i_1'$, or
3. $i_1 + i_2 + i_3 = i_1' + i_2' + i_3'$, $i_1 = i_1'$ with $i_2 > i_2'$.

The polynomial order corresponds to $s_{[i_1,i_2,i_3]} \prec s_{[i'_1,i'_2,i'_3]}$ defining in the same manner. Under this ordering, the leading monomial of $s_{[i_1,i_2,i_3]}(x_1,x_2,x_3)$ is $x_1^{i_1}x_2^{i_2}x_3^{i_3}$. The total number of shifts is

$$\omega = \sum_{i_1=0}^{m} \sum_{i_2=0}^{m-i_1} \sum_{i_3=0}^{m-i_1-i_2} 1 = \frac{(m+1)(m+2)(m+3)}{6}.$$

For instance, setting $m = 2$ yields a basis matrix B whose diagonal entries are exactly $X_1^{i_1}X_2^{i_2}X_3^{i_3}p^{m-i_2-i_3}$.

$$\left(\begin{array}{c|cccccccccc}
 & 1 & x_1 & x_2 & x_3 & x_1^2 & x_1x_2 & x_1x_3 & x_2^2 & x_2x_3 & x_3^2 \\ \hline
s_{[0,0,0]} & p^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
s_{[1,0,0]} & 0 & X_1p^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
s_{[0,1,0]} & -c_1p & rX_1p & X_2p & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
s_{[0,0,1]} & -c_2p & tX_1p & 0 & X_3p & 0 & 0 & 0 & 0 & 0 & 0 \\
s_{[2,0,0]} & 0 & 0 & 0 & 0 & X_1^2p^2 & 0 & 0 & 0 & 0 & 0 \\
s_{[1,1,0]} & 0 & -c_1X_1p & 0 & 0 & rX_1^2p & X_1X_2p & 0 & 0 & 0 & 0 \\
s_{[1,0,1]} & 0 & -c_2X_1p & 0 & 0 & tX_1^2p & 0 & X_1X_3p & 0 & 0 & 0 \\
s_{[0,2,0]} & c_1^2 & -2rc_1X_1 & -2c_1X_2 & 0 & r^2X_1^2 & 2rX_1X_2 & 0 & X_2^2 & 0 & 0 \\
s_{[0,1,1]} & c_1c_2 & -(rc_2+tc_1)X_1 & -c_2X_2 & -c_1X_3 & rtX_1^2 & tX_1X_2 & rX_1X_3 & 0 & X_2X_3 & 0 \\
s_{[0,0,2]} & c_2^2 & -2tc_2X_1 & 0 & -2c_2X_3 & t^2X_1^2 & 0 & 2tX_1X_3 & 0 & 0 & X_3^2
\end{array}\right).$$

In this setting, the lattice determinant is given by

$$\det(\Lambda) = X_1^{\sigma_1}X_2^{\sigma_2}X_3^{\sigma_3}p^{\sigma_p},$$

with the exponents computed as

$$\begin{aligned}
\sigma_1 &= \sum_{i_1=0}^{m} \sum_{i_2=0}^{m-i_1} \sum_{i_3=0}^{m-i_1-i_2} i_1 = \frac{m(m+1)(m+2)(m+3)}{24}, \\
\sigma_2 &= \sum_{i_1=0}^{m} \sum_{i_2=0}^{m-i_1} \sum_{i_3=0}^{m-i_1-i_2} i_2 = \frac{m(m+1)(m+2)(m+3)}{24}, \\
\sigma_3 &= \sum_{i_1=0}^{m} \sum_{i_2=0}^{m-i_1} \sum_{i_3=0}^{m-i_1-i_2} i_3 = \frac{m(m+1)(m+2)(m+3)}{24}, \\
\sigma_p &= \sum_{i_1=0}^{m} \sum_{i_2=0}^{m-i_1} \sum_{i_3=0}^{m-i_1-i_2} (m-i_2-i_3) = \frac{m(m+1)(m+2)(m+3)}{12}.
\end{aligned}$$

Enforcing the solving condition

$$\det(\Lambda) < R^{\omega}, \quad \text{with } R = p^m,$$

and substituting $X_i = p^{\xi_i}$ for $i = 1,2,3$, the inequality yields

$$X_1^{\frac{m(m+1)(m+2)(m+3)}{24}} X_2^{\frac{m(m+1)(m+2)(m+3)}{24}} X_2^{\frac{m(m+1)(m+2)(m+3)}{24}} p^{\frac{m(m+1)(m+2)(m+3)}{12}} < p^{m\cdot\frac{(m+1)(m+2)(m+3)}{6}},$$

which simplifies to

$$\frac{1}{24}(\xi_1+\xi_2+\xi_3)+\frac{1}{12}<\frac{1}{6}.$$

Thus, we obtain the bound

$$\xi_1+\xi_2+\xi_3<2. \tag{10}$$

Once several integer polynomials $v_k(x_1,x_2,x_3)$ are derived (each vanishing at the common root (a,b_1,b_2)), one can recover these unknowns via methods such as Gröbner method or resultant computations. The LLL reduction step dominates the overall running time, and since m is fixed, the complexity is polynomial in $\log p = n$. The former analysis is summarized in the following proposition.

Proposition 2. *Let $p=2^n-1$ be an n-bit Mersenne prime and let w be a positive integer. Suppose the unknown values a, b_1, and b_2 with Hamming weight w satisfy the bounds $a\le p^{\xi_1}$, $b_1\le p^{\xi_2}$, and $b_2\le p^{\xi_3}$. Given c_1, c_2, r, and t with $c_1=ar+b_1$ and $c_2=at+b_2$, a, b_1, and b_2 can be recovered in time polynomial in n provided that $\xi_1+\xi_2+\xi_3<2$.*

The condition $\xi_1+\xi_2+\xi_3<2$ in effect guarantees that $ab_1b_2<p^2$, which generalizes the earlier restriction in [11] where each unknown was assumed to be below $p^{2/3}$. This extension accommodates unbalanced parameters, thereby relaxing size constraints and widening the attack's range.

For the success probability analysis, we represent a, b_1, and b_2 as n-bit strings with w ones. In the previous attack of Coron and Gini [11], the probability of all unknowns being confined to the lower $\lfloor 2n/3\rfloor$ bits was roughly

$$\frac{\binom{\lfloor 2n/3\rfloor}{w}\binom{\lfloor 2n/3\rfloor}{w}\binom{\lfloor 2n/3\rfloor}{w}}{\binom{n}{w}\binom{n}{w}\binom{n}{w}}=\left(\frac{\lfloor 2n/3\rfloor!(n-w)!}{(\lfloor 2n/3\rfloor-w)!\,n!}\right)^3\approx\left(\frac{2}{3}\right)^{3w}\approx 2^{-1.75w}.$$

The success probability of our attack is denoted by Pr_3. One may loosely express it as

$$\mathrm{Pr}_3=\frac{\sum_{k_1=w}^{2n-2w}\sum_{k_2=w}^{2n-k_1-w}\binom{k_1}{w}\binom{k_2}{w}\binom{2n-k_1-k_2}{w}}{\binom{n}{w}\binom{n}{w}\binom{n}{w}}.$$

By applying the combinatorial identity

$$\sum_{k=i}^{n-j}\binom{k}{i}\binom{n-k}{j}=\binom{n+1}{i+j+1},$$

this simplifies to

$$\mathrm{Pr}_3=\frac{\sum_{k_1=w}^{2n-2w}\binom{k_1}{w}\binom{2n+1-k_1}{2w+1}}{\binom{n}{w}\binom{n}{w}\binom{n}{w}}=\frac{\binom{2n+2}{3w+2}}{\binom{n}{w}^3}.$$

Using Stirling's approximation and the assumption that $w\approx\sqrt{n}$ (i.e., $n\approx w^2$ for $n\to\infty$), an asymptotic analysis reveals that the improvement factor satisfies

$$\mathsf{r}_3=\frac{\binom{2n+2}{3w+2}}{\binom{\lfloor 2n/3\rfloor}{w}^3}=\frac{(2n+2)!(w!(\lfloor 2n/3\rfloor-w)!)^3}{(3w+2)!(2n-3w)!(\lfloor 2n/3\rfloor!)^3}\approx w^3.$$

More details can be found in Appendix A.2. Consequently, the overall success probability $\Pr_3$ of the above attack is approximately $\Pr_3 \approx w^3\, 2^{-1.75w}$.

Remark 1. Although both continued fraction and lattice-based methods leverage approximation of modular relations, they operate fundamentally differently. The former exploits one-dimensional rational structure, while the latter handles multivariate modular polynomial equations via basis reduction.

5 Validating Experiments

To demonstrate the correctness and effectiveness of our proposed attacks on *MLHRSP* and *MLHCSP* (as established in Theorems 2 and 3, Propositions 1 and 2), we performed a series of numerical experiments with randomly generated parameters. All experiments were carried out using `SageMath` [23] on a machine running 64-bit Windows 10 with Ubuntu 22.04 on WSL 2, equipped with a 2.80 GHz CPU and 16 GB of RAM. It is important to note that while we experimented with $w \in [30, 70]$ to verify the correctness of the final Hamming weight check, the core recovery phase is insensitive to w. The reported running times are primarily functions regarding n. Consequently, our proposed attacks remain computationally negligible (taking only seconds) even if w increases to $w = 128$, as long as the attack condition is satisfied.

For the first attack, we generated *MLHRSP* instances with parameters $p = 2^n - 1$, n, w, and ξ_1, ξ_2. Starting from two n-bit random integers f and g with Hamming weight w satisfying $fg < p$, we computed the public key h via the equation $h \equiv f/g \pmod{p}$. To test the key recovery process, we gradually increased the values of f and g, thereby achieving larger experimental bounds ξ_1 and ξ_2. For each experimental setting, five independent trials were performed, all yielding a 100% success rate. The results are summarized in Table 2.

Table 2. Experimental results of the first attack on *MLHRSP*

n	p	w	ξ_1	ξ_2	$\xi_1 + \xi_2$	Time (Sec.)
9689	$2^{9689} - 1$	45	0.500	0.500	1.000	1.80
			0.350	0.650	1.000	3.12
			0.150	0.850	1.000	5.36
11213	$2^{11213} - 1$	50	0.500	0.500	1.000	2.48
			0.200	0.800	1.000	6.69
			0.100	0.900	1.000	8.52
19937	$2^{19937} - 1$	70	0.500	0.500	1.000	8.31
			0.400	0.600	1.000	12.34
			0.300	0.700	1.000	17.29

In each experiment, we successfully found the desired convergent $\frac{k}{g}$ with the constraint that the Hamming weight of g is w. Then we computed $f = hg - kp$ and solved the *MLHRSP*, thus breaking the AJPS cryptosystem. As observed in Table 2, the running time increases with the size of the modulus p. Moreover,

experiments indicate that more unbalanced private key choices (i.e., $f < g$) lead to higher computational costs. In general, the continued fraction-based attacks require only a few seconds per trial.

Furthermore, our experiments efficiently recovered the private key (f, g) using only the given h and p, without requiring any prior knowledge or estimates of upper bounds on f and g (i.e, without knowing ξ_1 and ξ_2). This demonstrates a significant advantage of our new attack over previous approaches [6,26].

For the second attack, we generated *MLHCSP* instances with parameters $p = 2^n - 1$, n, w, and ξ_1, ξ_2. Starting from two n-bit random integers f and g with Hamming weight w satisfying $fg < p$, we computed the public key (r, t) via the equation $t \equiv rf + g \pmod{p}$, with r chosen at random. To test the key recovery process, we gradually increased the values of f and g, thereby achieving larger experimental bounds ξ_1 and ξ_2. For each experimental setting, five independent trials were performed, all yielding a 100% success rate. The results are summarized in Table 3.

Table 3. Experimental results of the second attack on *MLHCSP* (bit)

n	p	w	ξ_1	ξ_2	$\xi_1 + \xi_2$	Time (Sec.)
4253	$2^{4253} - 1$	30	0.897	0.102	0.999	23.86
			0.429	0.570	0.999	8.59
			0.063	0.936	0.999	0.95
9689	$2^{9689} - 1$	45	0.923	0.077	1.000	141.62
			0.524	0.476	1.000	64.34
			0.235	0.765	1.000	25.15
11213	$2^{11213} - 1$	50	0.500	0.500	1.000	84.31
			0.200	0.800	1.000	26.85
			0.100	0.900	1.000	12.88
19937	$2^{19937} - 1$	70	0.500	0.500	1.000	310.20
			0.400	0.600	1.000	229.51
			0.300	0.700	1.000	167.39

In each experiment, we successfully determined the desired a and b based on convergents from the continued fraction expansion of $\frac{r}{p}$. Thus, we derived f and g, solved *MLHCSP*, and broke the AJPS cryptosystem. As observed in Table 3, the running time increases with the size of the modulus p. Moreover, experiments indicate that more unbalanced choices using larger f lead to higher computational costs. In general, the continued fraction-based attacks require tens of seconds on average per trial. Furthermore, our experiments efficiently recovered the private key (f, g) using only the given r, t, and p, without requiring any prior knowledge or estimates of upper bounds on f and g.

For the third attack, we generated *MLHCSP* instances with parameters $p = 2^n - 1$, n, w, and ξ_1, ξ_2. Starting from two n-bit random integers f and g with Hamming weight w satisfying $fg < p$, we computed the public key (r, t) via the equation $t \equiv rf + g \pmod{p}$, with r chosen at random. To test the key recovery process, we gradually increased the values of f and g, thereby achieving larger

experimental bounds ξ_1 and ξ_2. For each setting, a suitable parameter m was selected to construct the lattice, and five independent trials were performed, all yielding a 100% success rate. The results are summarized in Table 4.

Table 4. Experimental results of the third attack on *MLHCSP* (bit)

n	p	w	ξ_1	ξ_2	$\xi_1+\xi_2$	m	ω	Time (Sec.)
9689	$2^{9689}-1$	45	0.500	0.500	1.000	3	10	0.47
			0.350	0.650	1.000	3	10	0.34
			0.150	0.850	1.000	3	10	0.16
11213	$2^{11213}-1$	50	0.500	0.500	1.000	3	10	1.61
			0.200	0.800	1.000	3	10	0.24
			0.100	0.900	1.000	3	10	0.15
19937	$2^{19937}-1$	70	0.500	0.500	1.000	3	10	4.70
			0.400	0.600	1.000	3	10	3.79
			0.300	0.700	1.000	3	10	2.93

In each experiment, we collected a sufficient number of polynomials that met the solvability criteria after applying the LLL algorithm. As seen in Table 4, the running time increases with the lattice dimension ω and the size of the modulus p, since both factors directly affect the dimensions and entries of the lattice basis matrix. Additionally, experiments indicate that more unbalanced private key choices (i.e., $f < g$) lead to lower computational costs. In general, both lattice reduction and subsequent root extraction require only a few seconds or less than one second per trial.

By converting the reduced lattice vectors into integer polynomials, we successfully extracted the common root $(x_1^\star, x_2^\star) = (f, g)$, thus solving the *MLHCSP* and breaking the AJPS cryptosystem. Moreover, our experiments conform to the theoretical bounds, even when a low-dimensional lattice is constructed.

For the fourth attack, *MLHCSP* instances were similarly generated using parameters $p = 2^n - 1$, n, w, and ξ_1, ξ_2, ξ_3. Starting from two random integers f, g with Hamming weight w and a random integer w, we computed the public key (r, t) via $t \equiv rf + g \pmod{p}$. Subsequently, random integers a, b_1, and b_2 of Hamming weight w, satisfying $ab_1b_2 < p^2$, were generated. The ciphertext components were then computed as $c_1 = ar + b_1$ and $c_2 = at + b_2$ (assuming the message $m = 0$). By gradually increasing a, b_1, and b_2, we achieved larger experimental bounds ξ_1, ξ_2, and ξ_3, which facilitated a successful ciphertext distinguishing attack. Similar to the second attack, a suitable parameter m was chosen for lattice construction, and five independent trials yielded a 100% success rate. The corresponding results for the *MLHCSP* KEM variant are shown in Table 5.

Similarly to the third attack, we collected a sufficient number of polynomials meeting the solvability condition after running the LLL algorithm. As shown in Table 5, the running time grows with the lattice dimension and the size of p, reflecting the increased complexity of the lattice basis. Once the reduced vectors were transformed into integer polynomials, we easily recovered the common root

Table 5. Experimental results of the fourth attack on *MLHCSP* (KEM)

n	w	ξ_1	ξ_2	ξ_3	$\xi_1+\xi_2+\xi_3$	m	ω	Time (Sec.)
4253	30	0.670	0.670	0.660	2.000	3	20	1.49
		0.350	0.650	1.000	2.000	3	20	0.91
		0.200	1.200	0.600	2.000	3	20	0.65
9689	45	0.660	0.670	0.670	2.000	3	20	16.52
		0.600	0.500	0.900	2.000	3	20	16.60
		0.650	0.950	0.400	2.000	3	20	13.41
11213	50	0.670	0.660	0.670	2.000	3	20	25.23
		0.300	0.700	1.000	2.000	3	20	3.55
		0.200	0.300	1.500	2.000	3	20	4.51
19937	70	0.670	0.670	0.660	2.000	3	20	59.44
		0.600	0.600	0.800	2.000	3	20	72.15
		0.500	0.850	0.650	2.000	3	20	54.46

$(x_1^\star, x_2^\star, x_3^\star) = (a, b_1, b_2)$, which allowed us to distinguish whether the ciphertext (and hence the plaintext) was zero or not.

The successful extraction of (a, b_1, b_2) demonstrates the effectiveness of our ciphertext distinguishing attack on *MLHCSP*. Furthermore, even with a small lattice dimension (e.g., $\omega = 20$), our experimental results are in full agreement with the theoretical bounds. Both lattice reduction and subsequent root extraction require an average of dozens of seconds per trial.

To provide further insight into the implementation of our proposed attacks, we include additional toy numerical examples in Appendix B. These examples briefly detail the intermediate steps, parameter selections, and the attack process, thereby validating our theoretical predictions through practical application. They also serve as a comprehensive resource for readers interested in replicating our experiments or exploring the practical aspects of the proposed attacks.

6 Concluding Remarks

In this work, we revisited the Mersenne number-based AJPS cryptosystem by examining its foundational hard problems, namely *MLHRSP* and *MLHCSP*. We introduce two novel continued fraction-based attacks alongside two enhanced lattice-based attacks, designed to target private key recovery and ciphertext indistinguishability. Specifically, the continued fraction-based attacks further relax size constraints by eliminating the need to estimate private key bounds, offering a significant improvement over previous attacks. Moreover, our numerical experiments substantiate the practical effectiveness of this approach.

Our new lattice-based attacks leverage a tailored solving strategy focusing on modular polynomial equations, yielding two main advancements. First, they expand the scope of vulnerable key scenarios, making the AJPS cryptosystem more susceptible to breaches. Second, they enhance the success probability in unbalanced attack settings, improving reliability under diverse conditions. These refinements strengthen the current attack framework and demonstrate its effectiveness.

A promising direction for future research involves transforming attack success probability into generic time complexity estimates by employing the random partition technique used in earlier work. Integrating this technique may enhance attack performance, potentially unlocking further improvements in efficiency and applicability.

References

1. Aggarwal, D., Joux, A., Prakash, A., Santha, M.: A new public-key cryptosystem via Mersenne numbers. Cryptology ePrint Archive, Paper 2017/481 (2017), https://eprint.iacr.org/2017/481
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 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part III. Lecture Notes in Computer Science, vol. 10993, pp. 459–482. Springer (2018), https://doi.org/10.1007/978-3-319-96878-0_16
3. Aggarwal, D., Joux, A., Prakash, A., Santha, M.: Mersenne-756839. Tech. rep., National Institute of Standards and Technology (2017), https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-1-submissions
4. Bailey, D.H., Borwein, J.M., Crandall, R.E.: On the Khintchine constant. Math. Comput. **66**(217), 417–431 (1997). https://doi.org/10.1090/S0025-5718-97-00800-4, https://doi.org/10.1090/S0025-5718-97-00800-4
5. Becker, T., Weispfenning, V., Kredel, H.: Gröbner bases - a computational approach to commutative algebra, Graduate texts in mathematics, vol. 141. Springer (1993)
6. Beunardeau, M., Connolly, A., Géraud, R., Naccache, D.: On the hardness of the Mersenne low hamming ratio assumption. In: Lange, T., Dunkelman, O. (eds.) Progress in Cryptology - LATINCRYPT 2017 - 5th International Conference on Cryptology and Information Security in Latin America, Havana, Cuba, September 20-22, 2017, Revised Selected Papers. Lecture Notes in Computer Science, vol. 11368, pp. 166–174. Springer (2017), https://doi.org/10.1007/978-3-030-25283-0_9
7. Boer, K., Ducas, L., Jeffery, S., Wolf, R.: Attacks on the AJPS Mersenne-based cryptosystem. In: Lange, T., Steinwandt, R. (eds.) Post-Quantum Cryptography - 9th International Conference, PQCrypto 2018, Fort Lauderdale, FL, USA, April 9-11, 2018, Proceedings. Lecture Notes in Computer Science, vol. 10786, pp. 101–120. Springer (2018), https://doi.org/10.1007/978-3-319-79063-3_5
8. Bootland, C., Castryck, W., Szepieniec, A., Vercauteren, F.: SoK: On the security of cryptographic problems from linear algebra. Mathematical Cryptology **3**(1), 52–95 (Jul 2023), https://journals.flvc.org/mathcryptology/article/view/129672
9. Budroni, A., Tenti, A.: The Mersenne low hamming combination search problem can be reduced to an ILP problem. In: Buchmann, J., Nitaj, A., Rachidi, T. (eds.) Progress in Cryptology - AFRICACRYPT 2019 - 11th International Conference on Cryptology in Africa, Rabat, Morocco, July 9-11, 2019, Proceedings. Lecture Notes in Computer Science, vol. 11627, pp. 41–55. Springer (2019), https://doi.org/10.1007/978-3-030-23696-0_3

10. Coppersmith, D.: Small solutions to polynomial equations, and low exponent RSA vulnerabilities. J. Cryptol. **10**(4), 233–260 (1997). https://doi.org/10.1007/s001459900030, https://doi.org/10.1007/s001459900030
11. Coron, J., Gini, A.: Improved cryptanalysis of the AJPS Mersenne based cryptosystem. J. Math. Cryptol. **14**(1), 218–223 (2020), https://doi.org/10.1515/jmc-2019-0027
12. Fatou, P.: Sur l'approximation des incommensurables et les séries trigonométriques. C. R. Acad. Sci., Paris **139**, 1019–1021 (1905)
13. Gallant, R.P., Lambert, R.J., Vanstone, S.A.: Faster point multiplication on elliptic curves with efficient endomorphisms. In: Kilian, J. (ed.) Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001, Proceedings. Lecture Notes in Computer Science, vol. 2139, pp. 190–200. Springer (2001). https://doi.org/10.1007/3-540-44647-8_11, https://doi.org/10.1007/3-540-44647-8_11
14. Grace, J.H.: The classification of rational approximations. Proc. Lond. Math. Soc. (2) **17**, 247–258 (1919). https://doi.org/10.1112/plms/s2-17.1.247, zenodo.org/record/1433509
15. Graham, R., Knuth, D., Patashnik, O.: Concrete Mathematics: A Foundation for Computer Science, 2nd Edtion. Addison-Wesley (1994), https://www-cs-faculty.stanford.edu/\%7Eknuth/gkp.html
16. Hoffstein, J., Pipher, J., Silverman, J.: NTRU: A ring-based public key cryptosystem. In: Buhler, J. (ed.) Algorithmic Number Theory, Third International Symposium, ANTS-III, Portland, Oregon, USA, June 21-25, 1998, Proceedings. Lecture Notes in Computer Science, vol. 1423, pp. 267–288. Springer (1998), https://doi.org/10.1007/BFb0054868
17. Howgrave-Graham, N.: Finding small roots of univariate modular equations revisited. In: Darnell, M. (ed.) Cryptography and Coding, 6th IMA International Conference, Cirencester, UK, December 17-19, 1997, Proceedings. Lecture Notes in Computer Science, vol. 1355, pp. 131–142. Springer (1997). https://doi.org/10.1007/BFb0024458, https://doi.org/10.1007/BFb0024458
18. Legendre, A.: Essai sur la théorie des nombres. Duprat, An VI, Paris (1798)
19. Lenstra, A., Lenstra, H., Lovász, L.: Factoring polynomials with rational coefficients. Mathematische Annalen **261**(4), 515–534 (1982)
20. May, A.: New RSA vulnerabilities using lattice reduction methods. Ph.D. thesis, University of Paderborn (2003), https://ubdata.uni-paderborn.de/ediss/17/2003/may/disserta.pdf
21. May, A.: Lattice-based integer factorization - an introduction to Coppersmith's method. In: Bos, J.W., Stam, M. (eds.) Computational Cryptography: Algorithmic Aspects of Cryptology, pp. 78–105. London Mathematical Society Lecture Note Series, Cambridge University Press (2021), https://www.cits.ruhr-uni-bochum.de/imperia/md/content/may/paper/intro_to_coppersmiths_method.pdf
22. Szepieniec, A.: Ramstake. Tech. rep., National Institute of Standards and Technology (2017), https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-1-submissions
23. The Sage Developers: SageMath, the Sage Mathematics Software System (Version 10.3) (2025), `https://www.sagemath.org`
24. Tiepelt, M., D'Anvers, J.: Exploiting decryption failures in Mersenne number cryptosystems. In: Emura, K., Yanai, N. (eds.) Proceedings of the 7th on ASIA Public-Key Cryptography Workshop, APKC@AsiaCCS 2020, Taipei, Taiwan, October 6, 2020. pp. 45–54. ACM (2020), https://doi.org/10.1145/3384940.3388957

25. Zheng, M., Kang, H.: Lattice-based cryptanalysis of RSA-type cryptosystems: a bibliometric analysis. Cybersecur. **7**(1), 74 (2024). https://doi.org/10.1186/S42400-024-00289-7, https://doi.org/10.1186/s42400-024-00289-7
26. Zheng, M., Yan, W.: Improved lattice-based attack on Mersenne low Hamming ratio search problem. In: Zhu, T., Li, Y. (eds.) Information Security and Privacy - 29th Australasian Conference, ACISP 2024, Sydney, NSW, Australia, July 15-17, 2024, Proceedings, Part II. Lecture Notes in Computer Science, vol. 14896, pp. 185–202. Springer (2024). https://doi.org/10.1007/978-981-97-5028-3_10, https://doi.org/10.1007/978-981-97-5028-3_10

A Success Probability Analysis

A.1 Type 2 Problem

Applying Stirling's approximation $k! \approx \sqrt{2\pi k}\left(\frac{k}{\mathrm{e}}\right)^k$, we obtain r_2 that is approximately

$$\frac{\sqrt{2\pi(n+1)}\left(\frac{n+1}{\mathrm{e}}\right)^{n+1}\cdot 2\pi w(\frac{w}{\mathrm{e}})^{2w}\cdot 2\pi(\lfloor n/2\rfloor - w)(\frac{\lfloor n/2\rfloor - w}{\mathrm{e}})^{2\lfloor n/2\rfloor - 2w}}{\sqrt{2\pi(2w+1)}\left(\frac{2w+1}{\mathrm{e}}\right)^{2w+1}\cdot\sqrt{2\pi(n-2w)}(\frac{n-2w}{\mathrm{e}})^{n-2w}\cdot 2\pi\lfloor n/2\rfloor(\frac{\lfloor n/2\rfloor}{\mathrm{e}})^{2\lfloor n/2\rfloor}}.$$

After canceling common terms and simplifying the dominating exponents, one finds that

$$\begin{aligned}
\mathsf{r}_2 &\approx \frac{(n+1)^{n+3/2}\cdot w^{2w+1}\cdot(\lfloor n/2\rfloor - w)^{2\lfloor n/2\rfloor - 2w+1}}{(2w+1)^{2w+3/2}\cdot(n-2w)^{n-2w+1/2}\cdot(\lfloor n/2\rfloor)^{2\lfloor n/2\rfloor+1}} \\
&\approx \frac{w^{2(w^2+3/2)}\cdot w^{2w+1}\cdot w^{2(w^2-2w+1)}}{w^{2w+3/2}\cdot w^{2(w^2-2w+1/2)}\cdot w^{2(w^2+1)}} \\
&= \frac{w^{2w^2+3+2w+1+2w^2-4w+2}}{w^{2w+3/2+2w^2-4w+1+2w^2+2}} \\
&= w^{(4w^2-2w+6)-(4w^2-2w+9/2)} \\
&= w^{3/2}.
\end{aligned}$$

Note that in this derivation, constant factors (e.g., those involving π) are omitted as they are negligible compared to the exponential terms.

A.2 Type 3 Problem

Similarly, we can approximately estimate r_3 as

$$\begin{aligned}
\mathsf{r}_3 &\approx \frac{\sqrt{2\pi(2n+2)}\left(\frac{2n+2}{\mathrm{e}}\right)^{2n+2}\cdot(2\pi w)^{3/2}\left(\frac{w}{\mathrm{e}}\right)^{3w}\cdot(2\pi(\lfloor 2n/3\rfloor - w))^{3/2}(\frac{\lfloor 2n/3\rfloor - w}{\mathrm{e}})^{3\lfloor 2n/3\rfloor - 3w}}{\sqrt{2\pi(3w+2)}\left(\frac{3w+2}{\mathrm{e}}\right)^{3w+2}\cdot\sqrt{2\pi(2n-3w)}\left(\frac{2n-3w}{\mathrm{e}}\right)^{2n-3w}\cdot(2\pi\lfloor 2n/3\rfloor)^{3/2}(\frac{\lfloor 2n/3\rfloor}{\mathrm{e}})^{3\lfloor 2n/3\rfloor}} \\
&\approx \frac{(n+1)^{2n+5/2}\, w^{3w+3/2}\, (\lfloor 2n/3\rfloor - w)^{3\lfloor 2n/3\rfloor - 3w+3/2}}{(3w+2)^{3w+5/2}\,(2n-3w)^{2n-3w+1/2}\,\lfloor 2n/3\rfloor^{3\lfloor 2n/3\rfloor+3/2}} \\
&\approx \frac{w^{2(2w^2+5/2)}\, w^{3w+3/2}\, w^{2(2w^2-3w+3/2)}}{w^{3w+5/2}\, w^{2(2w^2-3w+1/2)}\, w^{2(2w^2+3/2)}}
\end{aligned}$$

$$= w^{(8w^2-3w+19/2)-(8w^2-3w+13/2)}$$
$$= w^3.$$

In the derivation above, constant factors (e.g., those involving π) that are negligible compared to n have been omitted.

B Numerical Examples

B.1 The First Attack

We consider $n = 521$ (i.e., $p = 2^{521} - 1$) with $w = 23$, and h is given below,

$$h = 4073395231808991833689449778528583149337425218724276158093047519 5\backslash$$
$$2692712218120366771639167246226376173137631925632688619257271153 6\backslash$$
$$927675070787201736894596222.$$

We apply the attack strategy of Theorem 2. The convergent of the continued fraction expansion of $\frac{h}{p}$ for $i = 111$ is

$$p_{111} = 2044538992813890406857209323966677129260902615063610902,$$
$$q_{111} = 3445613719069804394052136490596583635248504604991085917.$$

Using $(f, g) = (p_{112} + p_{111}, q_{112} + q_{111})$, we get

$$k = 5470272013432838606006977863884671695742281980009705155,$$
$$g = 9218921411025159999083359591508264719070149455000928417.$$

Using $f = hg - kp$, we obtain

$$f = 5610298318312763561787132257583492980812218509523849469\backslash$$
$$3018284881962174952393493210849622574785632716 9.$$

which satisfy $h \equiv f/g \pmod{p}$ and the Hamming weight of f, g is exactly $w = 23$.

We observe that $f \approx p^{0.6488}$, $g \approx p^{0.3506}$, $fg \approx p^{0.9992}$, and $\frac{1}{2}p < fg < p$. This shows that the former methods for attacking *MLHRSP* in AJPS do not work.

B.2 The Second Attack

We consider $n = 521$ (i.e., $p = 2^{521} - 1$) with $w = 23$, and r, t are given below.

$$r = 4651711905140347594131937685456422626224870977420036403208 60\backslash$$
$$3621641245991426379623927821291552724202109721304286094379 87\backslash$$
$$525680306880730578487926576932128480 9$$
$$t = 1241468040579167078764833532893005179749907730996438644315 66\backslash$$
$$2698666078681228418059087152712009493528819452662985449604 27, \backslash$$
$$2212047739745838545308502969579578463.$$

We apply the attack strategy of Theorem 3. The convergents of the continued fraction expansion of $\frac{r}{p}$ for $i = 189$ and $i = 190$ are

$$\begin{aligned}
p_{189} &= 16343598288705310248171159917010968895834360540737634083755\backslash\\
&\quad 68207782969959087637607055360208158013,\\
q_{189} &= 24119183986101517069469282498149214220301560458848814182179\backslash\\
&\quad 10158305669767355656463131507290948888,\\
p_{190} &= 34733617861980213890992789829944445689578787063828247654757\backslash\\
&\quad 29529663845938369072516160654240445867,\\
q_{190} &= 51258389059586124913047155822309187555873403215700632839698\backslash\\
&\quad 97580354764686206026206271337363611069.
\end{aligned}$$

Moreover, the recovered integers a and b are

$$\begin{aligned}
a &= -92698510545521699406749725154339793889884009695327558768519\backslash\\
&\quad 05928173412687310172801488230865409 8,\\
b &= -43618468549332560217317475842171999484240209954611800342977\backslash\\
&\quad 13\backslash\\
&\quad 62130671419212288302530419242819166 1.
\end{aligned}$$

Using a and b, we obtain

$$\begin{aligned}
f &= 13673654522421766419190446229745836656325363486567934986500\backslash\\
&\quad 5\backslash\\
&\quad 65167872422926642065986209776400335258 5,\\
g &= 28246957857502444255990498253996820392623834081150655037961.
\end{aligned}$$

which satisfy $t \equiv rf + g \pmod{p}$ and their Hamming weight is exactly $w = 23$.

We observe that $f \approx p^{0.6259}$, $g \approx p^{0.3727}$, $fg \approx p^{0.9986}$, and $\frac{1}{2}p < fg < p$. This shows that the former methods for attacking *MLHCSP* in AJPS do not work.

B.3 The Third Attack

We consider $n = 521$ (i.e., $p = 2^{521} - 1$) with $w = 10$, and assume that two unbalanced integers f and g are less than 2^{130} and 2^{391}, respectively. We have

$$\begin{aligned}
r &= 46517119051403475941319376854564226262248709774200364032086\backslash\\
&\quad 0\backslash\\
&\quad 36216412459914263796239278212915527242021097213042860943798\backslash\\
&\quad 7\backslash\\
&\quad 52568030688073057848792657693212848 09,\\
t &= 66536602599930609276341957283199690687395529974566596272984 3\backslash\\
&\quad 64081630218978160732197387709315242921025495222062028943821 0\backslash\\
&\quad 99529282474649944018301567817380240 88.
\end{aligned}$$

We set $m = 3$ to construct a 10-dimensional lattice. After less than one second, we successfully extract the desired root $(x_1^\star, x_2^\star) = (f, g)$ as follows.

$$\begin{aligned}
x_1^\star &= 68056473637965582092433120544343405414 8,\\
x_2^\star &= 12608738179150423764138307185171200908870166920832135347353 1\backslash\\
&\quad 69925240023774561132515360190988227141606726932 1684373299 3.
\end{aligned}$$

B.4 The Fourth Attack

We consider $n = 521$ (i.e., $p = 2^{521} - 1$) with $w = 10$, and assume that two integers f and g with Hamming weight w are randomly chosen. We have

$$\begin{aligned}
r &= 11417137362040412705160475908020496359566344941249578118836 3\backslash\\
&\quad 35906384167750766707725994384586711740753173764605485612104 4\backslash\\
&\quad 14653240946067467321716386938441036 11,\\
t &= 67378209910993742694508176519355467252067424027638789580138 0\backslash\\
&\quad 60490083213430028203564725627099292746826617684064702327131 6\backslash\\
&\quad 78699299772275557798453653826985 27.
\end{aligned}$$

Besides, three unbalanced integers a, b_1 and b_2 are assumed less than 2^{104}, 2^{365} and 2^{573}, respectively. The corresponding values of c_1 and c_2 are as follows.

$$\begin{aligned}
c_1 &= 57429843877853757158968919157136354748876087165226567133335 3\backslash\\
&\quad 04520030380102377521942506104278127369334830529706927713465 3\backslash\\
&\quad 76201734042434840064968370811090078 79,\\
c_2 &= 27695718594386566944711195289252540557362567999172367235397 4\backslash\\
&\quad 08537746384315759686264712140569228506395696558457787693834 1\backslash\\
&\quad 52178474452537180904714833444074169 35.
\end{aligned}$$

We set $m = 3$ to construct a 20-dimensional lattice. After less than one second, we successfully extract the desired root $(x_1^\star, x_2^\star, x_3^\star) = (a, b_1, b_2)$ as follows.

$$\begin{aligned}
x_1^\star &= 10151108323028444308674214756352,\\
x_2^\star &= 21138101666104377275685613676584812168925076713990897216420 5\backslash\\
&\quad 29593107435079001387260353614552256547861550858240,\\
x_3^\star &= 15458150092069033378781407569971541746032964655375292535185 4\backslash\\
&\quad 21113833599934936704780241412520742105865297692075446932558 0\backslash\\
&\quad 05716750565661415707345453300150540255724538997768192.
\end{aligned}$$

It can be easily verified that all the parameters $p, r, t, a, b_1, b_2, c_1, c_2$ do satisfy the ciphertext generation regarding *MLHCSP*.