Article

An IND-CPA Analysis of a Cryptosystem Based on Bivariate Polynomial Reconstruction Problem

Siti Nabilah Yusof [1,†], Muhammad Rezal Kamel Ariffin [1,2,*,†], Terry Shue Chien Lau [3,†], Nur Raidah Salim [1,†], Sook-Chin Yip [4,*,†] and Timothy Tzen Vun Yap [3,†]

[1] Institute for Mathematical Research (INSPEM), Universiti Putra Malaysia, Serdang 43400, Selangor, Malaysia; sitinabilahyusof@gmail.com or gs53993@student.upm.edu.my (S.N.Y.); nurraidah@upm.edu.my (N.R.S.)
[2] Department of Mathematics and Statistics, Faculty of Science, Universiti Putra Malaysia, Serdang 43400, Selangor, Malaysia
[3] Faculty of Computing and Informatics, Multimedia University, Cyberjaya 63100, Selangor, Malaysia; terry.lau@mmu.edu.my (T.S.C.L.); timothy@mmu.edu.my (T.T.V.Y.)
[4] Faculty of Engineering, Multimedia University, Cyberjaya 63100, Selangor, Malaysia
* Correspondence: rezal@upm.edu.my (M.R.K.A.); scyip@mmu.edu.my (S.-C.Y.); Tel.: +60-3-97696838 (M.R.K.A.); +60-3-83125268 (S.-C.Y.)
† These authors contributed equally to this work.

Abstract: The Polynomial Reconstruction Problem (PRP) was introduced in 1999 as a new hard problem in post-quantum cryptography. Augot and Finiasz were the first to design a cryptographic system based on a univariate PRP, which was published at Eurocrypt 2003 and was broken in 2004. In 2013, a bivariate PRP was proposed. The design is a modified version of Augot and Finiasz's design. Our strategic method, comprising the modified Berlekamp–Welch algorithm and Coron strategies, allowed us to obtain certain secret parameters of the bivariate PRP. This finding resulted in us concluding that the bivariate PRP is not secure against Indistinguishable Chosen-Plaintext Attack (IND-CPA).

Keywords: Polynomial Reconstruction Problem; post-quantum cryptography; Indistinguishable Chosen-Plaintext Attack

MSC: 94A60; 11T71

check for updates

Citation: Yusof, S.N.; Kamel Ariffin, M.R.; Lau, T.S.C.; Salim, N.R.; Yip, S.-C.; Yap, T.T.V. An IND-CPA Analysis of a Cryptosystem Based on Bivariate Polynomial Reconstruction Problem. *Axioms* **2023**, *12*, 304. https://doi.org/10.3390/axioms12030304

Academic Editor: Alejandro R. Urzúa

Received: 10 December 2022
Revised: 24 February 2023
Accepted: 28 February 2023
Published: 17 March 2023

1. Introduction

The world of technology is evolving along with the last wall of defense of data security–cryptography. With the inevitable realization of the quantum computer, coupled with Shor's algorithm, which can solve the Integer Factorization Problem (IFP) and the Discrete Logarithm Problem (DLP) in polynomial time, classical cryptographic schemes depending on such hard mathematical problems could be vulnerable to quantum computer attack and rendered insecure. Among such cryptographic algorithms are the popular RSA and Elliptic Curve Cryptosystem (ECC) [1–4]. In 2016, the National Institute of Standards and Technology (NIST) had made a call for quantum resistant algorithms [4].

In quantum cryptography, a cryptographic algorithm is secure against the attack of both quantum and classical computers [5]. The popular Quantum Algorithm Zoo website lists favorable hard mathematical problems that are thought to be quantum resistant [6]. The post-quantum cryptography goal is to create schemes that can be resistant to a quantum computer [7]. Therefore, it is important for researchers to investigate different hard problems to create new cryptographic schemes that are secure against the attack of a quantum computer and to keep current communication practices protected [8].

The Polynomial Reconstruction Problem (PRP) is one of the listed problems in [6]. It has the full complexity needed against quantum computers of $\mathcal{O}(q)$, where q is a prime number of n-bits. The PRP was introduced in 1999 as a potential hard mathematical problem for cryptographic design. When compared to the Reed–Solomon error correcting codes, the PRP has some similarity related to its formulation [9]. Furthermore, the PRP has been broadly studied from the point of view of solvability and robustness. Among the reasons why the PRP is recommended as a hard mathematical problem, as mentioned in [10] is, firstly, some evidence that shows that the PRP can cope with the improvement of quantum computing. Secondly, this system has new advantages from the perspective of efficiency and cost effectiveness. Thirdly, the PRP uses simple matrix operations and other interesting components that might make it useful in cryptographic settings.

The PRP can be solved in polynomial time when the weight of error w is small enough, such that $w \leq \frac{n-k}{2}$, where n is the number of elements in a vector and k is the degree of the polynomial. Guruswami and Sudan improved this to $w \leq n - \sqrt{kn}$ [11]. In 2003, Augot and Finiasz proposed a cryptosystem that utilizes the PRP [12]. We denote this cryptosystem as the AF-Cryptosystem. The AF-Cryptosystem utilizes two types of PRPs. The first PRP concerns the definition in [6]. The second PRP is a specially constructed PRP to ensure decryption. The second PRP, which we coin as the Augot and Finiasz Solvable PRP (AF-SPRP) is defined below.

Definition 1. ***(Augot and Finiasz Solvable PRP)*** *Given n, k, t and $(x_i, y_i)_{i=1,\cdots,n}$, output any polynomial p such that deg $< k$ and $p(x_i) = y_i$ for at least t values of i, where $t = n - w$.*

The AF-Cryptosystem utilizes a univariate polynomial [13,14]. The AF-SPRP as in Definition 1 ensures that decryption can occur. That is, when one is given t points on a Cartesian plane, one needs to output a polynomial that fits all the points. Parameter t represents the number of elements equal to 0 in the vector. To complete the decryption process, Lagrange interpolation is utilized.

In 2004, the AF-Cryptosystem was successfully cryptanalyzed by Coron, where Coron managed to obtain the plaintext in polynomial time [15]. Nevertheless, the idea to utilize a PRP for a cryptosystem is indeed tempting. In 2013, Ajeena et al. utilized bivariate polynomials and the Vandermonde matrix to put forward a new PRP-based cryptosystem [16]. We denote this cryptosystem as the AAK-Cryptosystem. The designers of the AAK-Cryptosystem claimed that increasing the number of variables increases the level of security and resistance against any attack.

Designers of cryptosystems usually claim the security of the design in terms of exponential time and memory needed for the attack [17]. It is an essential characteristic to verify the security of a cryptographic scheme [18]. At the same time, it must be noted that indistinguishability is also an essential characteristic for a cryptosystem that might be chosen to be used on a plaintext domain of non-exponential size. A design needs to be secure against Indistinguishable Chosen-Plaintext Attack (i.e., IND-CPA secure) in order to overcome an adversary having the capability to re-encrypt all possible plaintexts and make a comparison with the ciphertext.

A cryptosystem is IND-CPA secure if every Probabilistic Polynomial Time Adversary has a negligible "advantage" over random guessing. An IND-CPA-secure cryptosystem results in an adversary not being able to win the IND-CPA game with probability more than $\frac{1}{2} + \varepsilon(n)$, where $\varepsilon(n)$ is a negligible function in security parameter n. To this end, this research on the AAK-Cryptosystem is to determine whether it is IND-CPA secure or not.

Our contribution: This paper puts forward an IND-CPA analysis of the AAK-Cryptosystem that is the extension of [19]. The motivation for this research originates from the cryptanalysis performed on the AF-Cryptosystem by Coron. We used the Berlekamp–Welch algorithm and created a modified Coron cryptanalysis strategy, and we prove that we can construct a list of possible candidates of the AAK-Cryptosystem secret key, α. As such, we can highlight that the AAK-Cryptosystem is not IND-CPA secure.

The outline of this paper is shown as follows: In Section 2, we describe fundamental knowledge about the PRP as well as the Vandermonde method and outline the AAK-

Cryptosystem. We also put forward the definition of Indistinguishable under Chosen-Plaintext Attack (IND-CPA). Next, we describe our proposed attack on the AAK-Cryptosystem and provide a numerical illustration for this attack in Section 3. Finally, we conclude in Section 4.

2. Materials and Methods

This section presents the fundamentals of PRP, Vandermonde method, AAK-Cryptosystem and IND-CPA concept.

2.1. Polynomial Reconstruction Problem (PRP)

We begin by revising fundamental knowledge regarding the PRP. The PRP has been well known since the generalized Reed–Solomon list decoding problem was reduced to it [20,21]. The PRP also seems to be hard, which results in it being a potential source of a hard mathematical problems to establish a cryptosystem [22]. To fathom the PRP, we here put forward the definition of PRP sourced from [6].

Definition 2. ***(Polynomial Reconstruction Problem from Quantum Zoo)*** *Let $p(x) = a_k x^k + \cdots + a_1 x + a_0$ be a polynomial over finite field $\mathbb{F}_q$. One is given access to the oracle and query value of $x_i \in \mathbb{F}_q$, where $1 \le i \le k+1$ and then outputs coefficients $a_k, \ldots, a_0$ to determine $p(x)$.*

When an oracle receives input $x \in \mathbb{F}_q$, it outputs $p(x)$. The objective of solving the PRP is to obtain coefficients $a_k, \ldots, a_0$ [6]. Note that the value of k is unknown and input x is less than q. Classically, the queries that are required to determine the coefficients are $k+1$. In the case of univariate polynomials of degree k, the PRP has query complexity of $\mathcal{O}\binom{k+1}{k}$.

2.2. PRP Computational Complexity

The highest degree for $p(x)$ is k and the number of coefficients in $p(x)$ is $k+1 = q-1$; this means that $k = q-2$. Therefore,

$$\mathcal{O}\binom{k+1}{k} = \mathcal{O}(q-1).$$

If $q \approx 2^n$ is exponentially large, it is impossible to query input x up to 2^n times. Hence, solving the PRP takes exponential time, which is $\mathcal{O}(2^n)$.

2.3. Vandermonde Method

The Vandermonde method is a method that is used to find an interpolating polynomial in two or more dimensions. Let us suppose that we have two dimensional points, $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ and that we obtain polynomial values for each point, denoted by $z_1, z_2, \ldots, z_n$, respectively. We want to find a bivariate polynomial of degree $n-1$ that fits all of these points. The step-by-step method is as follows:

1. Write the general formula of the bivariate polynomial of degree $n-1$.
2. Evaluate the polynomial at points $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$.
3. Solve the linear equation system.

The problem can easily be written as $V \cdot c = Z$, where Z is the vector of z values and c is the coefficient vector. This method is utilized in the decryption process of the AAK-Cryptosystem.

2.4. AAK-Cryptosystem

Ajeena et al. [16] proposed a bivariate PRP cryptosystem as described below. Let n be the number of elements in the vector. The AAK-Cryptosystem takes into consideration the below parameters (Table 1).

Table 1. Parameters used in the AAK-Cryptosystem.

Parameter	Remark
X	Input x_i
Y	Input y_i
$\mathbb{F}_q$	Finite field with size q
n	The number of elements in a vector
k	Its dimension
W	The weight of big error vector E when the PRP is hard, that is, $W > \frac{n-k}{2}$ [16]
w	The weight of small error e, which results in the PRP being able to decrypt the ciphertext such that $w \leq \frac{n-k}{2}$ [15]

Remark 1. *Value w represents the maximum number of elements not equal to 0 in a vector.*

Remark 2. *Value $n - w$ represents the number of elements equal to 0 in a vector.*

Utilizing the above parameters, ref. [16] constructed their cryptosystem with Algorithms 1–3.

```
Algorithm 1 Key Generation Process
Input: Parameters $(x_i, y_i, q, n, k, W, w)$
Output: Public Key, $PK$ and secret key pair $(C, E)$
1. Alice secretly generates monic bivariate polynomial $p(X, Y)$ of degree equal to $k - 1$ with respect to $X$ and $Y$ and big error vector $E$ with the weight of $W$.
2. Alice computes codeword $C = ev(p(X, Y)) = p(x_i, y_i)$ where $x_i, y_i \in \mathbb{F}_q$ and computes $PK = C + E$.
3. Output public key, $PK$ secret key pair $(C, E)$.
```

```
Algorithm 2 Encryption Process
Input: Message, $\mu \in \mathbb{F}_q$
Output: Ciphertext, $CT$
1. Bob wants to send a message polynomial $\mu(X, Y)$ with length $k + 1$.
2. The message is encoded into a codeword $\mu$ by computing $\mu = ev(\mu(X, Y)) = \mu(x_i, y_i)$.
3. Bob randomly generates $\alpha \in \mathbb{F}_q$ and small error vector $e$ with the weight of $w$.
4. Bob computes ciphertext $CT = \mu + \alpha \times PK + e$ and sends the ciphertext to Alice.
```

```
Algorithm 3 Decryption Process
Input: Ciphertext, $CT$
Output: Message polynomial, $\mu(X, Y)$
1. For $i$, where $E_i = 0$, determine $\overline{CT} = \overline{\mu} + \alpha \times \overline{C} + \overline{e}$.
2. Correct $\overline{CT}$ to obtain $\tilde{C}T = \tilde{\mu} + \alpha \times \tilde{C}$.
3. Compute unique polynomial $q(X, Y)$ of degree $k - 1$ by using Vandermonde method.
4. Determine the leading coefficient $q(X, Y)$.
5. Compute $\mu(X, Y) = q(X, Y) - \alpha p(X, Y)$.
```

Proof of Correctness

Proposition 1. *The AAK-Cryptosystem decryption algorithm is correct.*

Proof of Proposition 1. To show that from ciphertext CT, message $\mu(x, y)$ can be obtained, let us observe the following:

$$\begin{aligned} CT &= \mu + \alpha \times PK + e \\ &= \mu + \alpha \times (C + E) + e. \end{aligned} \tag{1}$$

Let us consider position $E_i = 0$. Let $\overline{\mu}$, $\overline{C}$, $\overline{e}$ and $\overline{CT}$ correspond to shortened codes μ, C, e and CT, respectively. Now, (1) becomes

$$\overline{CT} = \overline{\mu} + \alpha \times \overline{C} + \overline{e}. \tag{2}$$

By (2), $\overline{\mu} + \alpha \times \overline{C} \in \overline{RS_k}$. Provided that e has weight that is less than error correction capacity $\overline{RS_k}$, then $\overline{CT}$ can be corrected and $\tilde{\mu} + \alpha \times \tilde{C}$ can be found. Using the Vandermonde method, we compute the unique polynomial $q(x, y)$ degree $k - 1$ and

$$ev(q(x_i, y_i)) = \tilde{\mu}_i + \alpha \times \tilde{C}_i \tag{3}$$

for $i \in \{1, 2, \ldots, n\}$. Since we know that $ev(q(x_i, y_i)) = q(x_i, y_i)$, $\tilde{C} = ev(p(x_i, y_i)) = p(x_i, y_i)$ and $\tilde{\mu} = ev(\mu(x_i, y_i)) = \mu(x_i, y_i)$,

$$\begin{aligned} q(x_i, y_i) &= \mu(x_i, y_i) + \alpha p(x_i, y_i) \\ \mu(x_i, y_i) &= q(x_i, y_i) - \alpha p(x_i, y_i). \end{aligned} \tag{4}$$

With (4), message $\mu(x, y)$ is obtained. □

2.5. Indistinguishable under Chosen-Plaintext Attack (IND-CPA)

Every cryptosystem needs to have its basic security requirements analyzed, especially its indistinguishability characteristics, in order to avoid any attack to the cryptographic protocol [23]. Indistinguishable under Chosen-Plaintext Attack (IND-CPA) is a security notion for cryptosystems where a Probabilistic Polynomial Time Adversary (PPTA) communicates with a random oracle in a two-phase session, i.e., the learning and challenge phases [24]. IND-CPA is defined below.

Definition 3. ***(Indistinguishable under Chosen-Plaintext Attack)*** *The IND-CPA security model is defined by the following game between random oracle and PPTA:*

1. *The random oracle initializes a cryptographic scheme and generates $(PK, SK) = Gen(1^n)$ as well as choosing random $b \in \{0, 1\}$ and publishing public key PK, while secret key SK is kept secret.*
2. *The PPTA chooses two messages, μ_0 and μ_1, and sends them to the random oracle.*
3. *The random oracle randomly chooses one out of the two messages and encrypts it; then, it sends ciphertext $C = enc(\mu_b, PK)$ to the PPTA.*
4. *The PPTA determines b'. If $b' = b$, then it outputs 1; else, 0.*

A cryptosystem is Indistinguishable under Chosen-Plaintext Attack if for any PPTA, there exists a negligible function $\varepsilon(n)$ such that

$$Pr(b' = b) \leq \frac{1}{2} + \varepsilon(n).$$

In other words, an IND-CPA-secure cryptosystem is a cryptosystem where any passive adversary that can eavesdrop in a communication between two parties cannot obtain any information about the encrypted message [25].

3. The Attack

In this section, we prove that the AAK-Cryptosystem is not IND-CPA secure. We also provide a numerical illustration.

3.1. Cryptanalysis of AAK-Cryptosystem

Theorem 1. *Let μ and e be as described in the AAK-Cryptosystem. If the adversary can correctly ascertain value $\mu + e$, then given public key PK and ciphertext CT, the adversary can recover secret key α in polynomial time.*

Proof of Theorem 1. Let CT_i, PK_i and e_i be vector elements in CT, PK and e, respectively. Let us recall that the vectors of ciphertext CT and public key PK are given by

$$CT_i = \mu(x_i, y_i) + \alpha \cdot PK_i + e_i \quad \forall\, 1 \leq i \leq n$$

and

$$PK_i = C_i + E_i \quad \forall\, 1 \leq i \leq n.$$

We know that vector C is from the evaluation of polynomial $p(x_i, y_i)$. Based on [16], polynomial $p(x_i, y_i)$ is a monic polynomial, and the highest power for this polynomial is up to $k-1$ with respect to both x and y. In addition, polynomial $\mu(x_i, y_i)$ must be of length $k+1$. Consider the following set of equations:

$$\exists\, V, \mu, \alpha \begin{cases} \deg(V) \leq k-1, \quad V \neq 0 \\ \forall i,\ V(x_i, y_i) \cdot (CT_i - \alpha \times PK_i) = V(x_i, y_i) \cdot \mu(x_i, y_i) \end{cases} \tag{5}$$

$$\exists\, V, N, \lambda \begin{cases} \deg(V) \leq k-1, \quad V \neq 0, \quad \deg(N) \leq k-1 \\ \forall i,\ V(x_i, y_i) \cdot (CT_i - \lambda \times PK_i) = N(x_i, y_i) \end{cases} \tag{6}$$

From here, we can see that any solution (5) gives a solution to (6), where one takes $\lambda = \alpha$ and $N(x_i, y_i) = \mu(x_i, y_i) \cdot V(x_i, y_i)$. For a given λ, Equation (6) gives $2k^2$ unknowns, which are the coefficients of polynomials $V(x_i, y_i)$ and $N(x_i, y_i)$, where

$$V(x_i, y_i) = v_k x_i^{k-1} y_i^{k-1} + \ldots + v_3 x_i y_i + v_2 x_i + v_1 y_i + v_0$$

$$N(x_i, y_i) = n_k x_i^{k-1} y_i^{k-1} + \ldots + n_3 x_i y_i + n_2 x_i + n_1 y_i + n_0$$

and Y is the vector of coordinates

$$Y = (v_0, \ldots, v_{k-1}, n_0, \ldots, n_{k-1}).$$

Next, a matrix $M(\lambda)$ is created with the following entries:

$$M(\lambda)_{i,a,b} = (CT_i - \lambda \cdot PK_i) \cdot (x_i)^a \cdot (y_i)^b \tag{7}$$

and

$$M(\lambda)_{i,a,b} = -(x_i)^a \cdot (y_i)^b \tag{8}$$

where $i \in \{1, \ldots, n\}$, $a \in \{0, \ldots, k-1\}$ and $b \in \{0, \ldots, k-1\}$ in (7) and (8). For the first half of columns of $M(\lambda)$, (7) is used, where a and b are the exponents of each monomial from polynomial $p(x, y)$. For the other half of columns of $M(\lambda)$, (8) is used, where a and b are also the exponents of each monomial from polynomial $p(x, y)$. Hence, $M(\lambda)$ is either a rectangular matrix or a square matrix.

Then, we consider $M(\lambda)$ with $\lambda = 0$ and use Gaussian elimination to compute the rank of matrix $M(0)$. Let us suppose that $M(\lambda)$ has dimensions $r \times s$. For rectangular matrix $M(\lambda)$, there are two cases:

(i) When $r > s$, if rank $M(0) = s$, then there exists sub-square matrix $M'(\lambda)$ in $M(\lambda)$.

(ii) When $r < s$, if rank $M(0) = r$, then there exists sub-square matrix $M'(\lambda)$ in $M(\lambda)$.

Using Equations (7) and (8), and with numerical input of public values (x_i, y_i), we create $M(\lambda)$, where λ represents the possible value of α. By (7), which we use in the first half of columns of $M(\lambda)$, we have

$$\begin{aligned}
M(\lambda)_{1,0,0} &= (CT_1 - \lambda \cdot PK_1) \cdot (x_1)^0 \cdot (y_1)^0 = (CT_1 - \lambda \cdot PK_1) \\
M(\lambda)_{1,1,0} &= (CT_1 - \lambda \cdot PK_1) \cdot (x_1)^1 \cdot (y_1)^0 = (CT_1 - \lambda \cdot PK_1) \cdot (x_1) \\
M(\lambda)_{1,0,1} &= (CT_1 - \lambda \cdot PK_1) \cdot (x_0)^1 \cdot (y_1)^1 = (CT_1 - \lambda \cdot PK_1) \cdot (x_1) \cdot (y_1) \\
&\vdots \\
M(\lambda)_{n,k-1,k-1} &= (CT_n - \lambda \cdot PK_n) \cdot (x_n)^{k-1} \cdot (y_n)^{k-1}
\end{aligned}$$

By (8), which we use in the second half of columns of $M(\lambda)$, we have

$$\begin{aligned}
M(\lambda)_{1,0,0} &= -(x_1)^0 \cdot (y_1)^0 = -1 \pmod q \\
M(\lambda)_{1,1,0} &= -(x_1)^1 \cdot (y_1)^0 = -x_1 \pmod q \\
M(\lambda)_{1,0,1} &= -(x_0)^1 \cdot (y_1)^1 = -y_1 \pmod q \\
&\vdots \\
M(\lambda)_{n,k-1,k-1} &= -(x_n)^{k-1} \cdot (y_n)^{k-1} \pmod q
\end{aligned}$$

When we put these equations into matrix $M(\lambda)$, we have

$$M(\lambda) = \begin{bmatrix}
(CT_1 - \lambda \cdot PK_1) & (CT_1 - \lambda \cdot PK_1) \cdot (x_1) & \cdots & (CT_1 - \lambda \cdot PK_1) \cdot (x_1)^{k-1} \cdot (y_1)^{k-1} & -1 & -x_1 & -y_1 & \cdots & -(x_1)^{k-1} \cdot (y_1)^{k-1} \\
(CT_2 - \lambda \cdot PK_2) & (CT_2 - \lambda \cdot PK_2) \cdot (x_2) & \cdots & (CT_2 - \lambda \cdot PK_2) \cdot (x_2)^{k-1} \cdot (y_2)^{k-1} & -1 & -x_2 & -y_2 & \cdots & -(x_2)^{k-1} \cdot (y_2)^{k-1} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
(CT_n - \lambda \cdot PK_n) & (CT_n - \lambda \cdot PK_n) \cdot (x_n) & \cdots & (CT_n - \lambda \cdot PK_n) \cdot (x_n)^{k-1} \cdot (y_n)^{k-1} & -1 & -x_n & -y_n & \cdots & -(x_n)^{k-1} \cdot (y_n)^{k-1}
\end{bmatrix}.$$

When Equations (7) and (8) are multiplied by $V(x_i, y_i)$ and $N(x_i, y_i)$, respectively, we have

$$V(x_i, y_i) \cdot M(\lambda)_{i,a,b} = (CT_i - \lambda \cdot PK_i) \cdot (x_i)^a \cdot (y_i)^b \cdot V(x_i, y_i) \tag{9}$$

and

$$N(x_i, y_i) \cdot M(\lambda)_{i,a,b} = -(x_i)^a \cdot (y_i)^b \cdot N(x_i, y_i). \tag{10}$$

The summation of (10) and (11) is

$$(CT_i - \lambda \cdot PK_i) \cdot (x_i)^a \cdot (y_i)^b \cdot V(x_i, y_i) - (x_i)^a \cdot (y_i)^b \cdot N(x_i, y_i). \tag{11}$$

Since $N(x_i, y_i) = \mu(x_i, y_i) \cdot V(x_i, y_i)$, (11) becomes

$$(CT_i - \lambda \cdot PK_i) \cdot (x_i)^a \cdot (y_i)^b \cdot V(x_i, y_i) - (x_i)^a \cdot (y_i)^b \cdot \mu(x_i, y_i) \cdot V(x_i, y_i). \tag{12}$$

Equation (5) shows that $V(x_i, y_i) \cdot (CT_i - \alpha \times PK_i) = V(x_i, y_i) \cdot \mu(x_i, y_i)$ and $\lambda = \alpha$; hence,

$$(x_i)^a \cdot (y_i)^b \cdot \mu(x_i, y_i) \cdot V(x_i, y_i) - (x_i)^a \cdot (y_i)^b \cdot \mu(x_i, y_i) \cdot V(x_i, y_i) = 0. \tag{13}$$

As such, Y contains the coefficients of polynomials $V(x_i, y_i)$ and $N(x_i, y_i)$, and if $\lambda = \alpha$, there exists Y such that

$$M(\lambda) \cdot Y = 0, \quad Y \neq 0. \tag{14}$$

If $M(\lambda)$ is a square matrix and rank $M(0) = r = s$, then we take $M(\lambda)$ as $M'(\lambda)$ to compute $f(\lambda) = \mathrm{Det}(M'(\lambda))$. If $M(\lambda)$ is a rectangular matrix, then we need to follow cases (i) and (ii) to find sub-square matrix $M'(\lambda)$. Sub-square matrix $M'(0)$ is invertible when the determinant is not equal to 0. Next, we need to identify parameter λ from matrix $M'(\lambda)$, which is constructed with relations (7) and (8). Given relation

$$M'(\lambda) \cdot Y = 0 \pmod{q} \tag{15}$$

the chosen rows or columns in $M(\lambda)$ can be arbitrary as long as the summation of (10) and (11) equals 0. Equation (15) shows a column matrix Y with entries not all equal to 0; then, Y corresponds to the nullspace of $M'(\lambda)$. This means that $M'(\lambda)$ is non-invertible and its determinant is equal to 0. As such, λ can be determined from relation $\mathrm{Det}(M'(\lambda)) = 0$. Hence, a solution of α must be a root for polynomial

$$f(\lambda) = \mathrm{Det}(M'(\lambda)).$$

To this end, the degree of polynomial $f(\lambda)$ is directly related to the number of columns containing λ in $M'(\lambda)$. The maximum number of columns possible is given by relation $\frac{n}{2}$. Note that n refers to the number of elements in the ciphertext (CT) vector. Let us observe that in order for the AAK-Cryptosystem to be practical, the number of elements in a vector cannot be exponentially large. Thus, the maximum number of roots is not exponentially large. Hence, if the adversary knows value $\mu + e$, the adversary can test all possible values of α in polynomial time. □

3.2. Algorithm for Theorem 1

The Algorithm 4 for Theorem 1 is shown below.

Algorithm 4 Listing all possible candidates of secret key α via Theorem 1

```
Input: Public key, $PK$ and ciphertext, $CT$
Output: Secret key, $\alpha$
1.  Compute public key, $PK = C + E$.
2.  Compute ciphertext, $CT = \mu + \alpha \times PK + e$.
3.  Construct matrix $M(\lambda)$:
4.    Compute first half column using $M(\lambda)_{i,a,b} = (CT_i - \lambda \cdot PK_i) \cdot (x_i)^a \cdot (y_i)^b$.
5.    Compute second half column using $M(\lambda)_{i,a,b} = -(x_i)^a \cdot (y_i)^b$.
6.  Get $[m, n] = M(\lambda)$ where $m$ represents as number of rows and $n$ represents as number
    of columns for matrix $M(\lambda)$.
7.  Apply $\alpha = 0$ to compute rank $M(0)$.
8.  Do the following procedure:
9.    if rank $M(0) = m = n$ then take $M(\lambda)$ as $M'(\lambda)$ end if
10.   if rank $M(0) = n$ then there exist sub square matrix $M'(\lambda)$ in $M(\lambda)$ end if
11.   if rank $M(0) = m$, then there exist sub square matrix $M'(\lambda)$ in $M(\lambda)$ end if
12. for sub square $M'(\lambda)$ do
13.   Compute determinant, $\mathrm{Det}(M'(\lambda))$.
14.   Solve $f(\lambda) = \mathrm{Det}(M'(\lambda)) = 0$.
15.   List all roots of $f(\lambda)$. This list contains all possible candidates of the secret key $\alpha$.
```

3.3. Numerical Illustration of Theorem 1

This section presents a numerical illustration of how to retrieve secret key α based on Theorem 1. Given $n = 10$, $k = 3$, $w = 1$ and $W = 3$ in $\mathbb{F}_{11}$, let $x = (2, 3, 3, 4, 5, 6, 7, 8, 9, 10)$ and $y = (4, 3, 6, 2, 1, 5, 7, 8, 9, 10)$. We take private polynomial

$$p(x, y) = x^2y + xy^2 + 3xy + 5$$

and big error vector E,

$$E = (0, 0, 0, 10, 0, 7, 3, 0, 0, 0).$$

The public key is

$$PK = C + E$$

where $C = ev(p(x,y))$. We compute C as follows:

$$p(2,4) = 0,\ p(3,3) = 9,\ p(3,6) = 1,\ p(4,2) = 0,\ p(5,1) = 6,$$

$$p(6,5) = 7,\ p(7,7) = 2,\ p(8,8) = 0,\ p(9,9) = 1,\ p(10,10) = 6.$$

Therefore,

$$\begin{aligned} PK &= C + E \\ &= (0,9,1,0,6,7,2,0,1,6) + (0,0,0,10,0,7,3,0,0,0) \\ &= (0,9,1,10,6,3,5,0,1,6). \end{aligned}$$

A message $\mu(x,y) = xy + 2x + 4y + 3$ is encoded into codeword μ, where $\mu = ev(m(x,y))$. That is,

$$\mu(2,4) = 9,\ \mu(3,3) = 8,\ \mu(3,6) = 7,\ \mu(4,2) = 5,\ \mu(5,1) = 0$$

$$\mu(6,5) = 10,\ \mu(7,7) = 6,\ \mu(8,8) = 5,\ \mu(9,9) = 6,\ \mu(10,10) = 9.$$

Therefore, we have

$$\mu = (9,8,7,5,0,10,6,5,6,9). \tag{16}$$

We choose private constant $\alpha = 3 \in \mathbb{F}_{11}$ and small error vector e, where

$$e = (0,0,0,0,0,7,0,0,0,0) \tag{17}$$

of weight $w = 1$. Ciphertext CT is

$$\begin{aligned} CT &= \mu + \alpha \times PK + e \\ &= (9,8,7,5,0,10,6,5,6,9) + 3 \times (0,9,1,10,6,3,5,0,1,6) + (0,0,0,0,0,7,0,0,0,0) \\ &= (9,8,7,5,0,10,6,5,6,9) + (0,5,3,8,7,9,4,0,3,7) + (0,0,0,0,0,7,0,0,0,0) \\ &= (9,2,10,2,7,4,10,5,9,5). \end{aligned}$$

We now proceed to attack ciphertext CT. Let $M(\lambda)$ be the matrix of the following system:

1. $M(\lambda)_{i,a,b} = (CT_i - \lambda \cdot PK_i) \cdot (x_i)^a \cdot (y_i)^b$
2. $M(\lambda)_{i,a,b} = -(x_i)^a \cdot (y_i)^b$

where $i \in \{1,\dots,10\}$, $a \in \{0,1,2\}$ and $b \in \{0,1,2\}$ in (1) and (2). For the first half of columns of matrix $M(\lambda)$, we use (1). Hence, when $i = 1$, $a = 0$ and $b = 0$,

$$\begin{aligned} M(\lambda)_{1,0,0} &= (CT_1 - \lambda \cdot PK_1) \cdot (x_1)^0 \cdot (y_1)^0 \\ &= 9 - \lambda \cdot 0 \\ &= 9. \end{aligned}$$

When $i = 5$, $a = 1$ and $b = 1$,

$$\begin{aligned} M(\lambda)_{5,1,1} &= (CT_5 - \lambda \cdot PK_5) \cdot (x_5)^1 \cdot (y_5)^1 \\ &= (7 - \lambda \cdot 6) \cdot 5 \cdot 1 \\ &= 2 - 8\lambda. \end{aligned}$$

When $i = 5$, $a = 2$ and $b = 1$,

$$\begin{aligned} M(\lambda)_{5,2,1} &= (CT_5 - \lambda \cdot PK_5) \cdot (x_5)^2 \cdot (y_5)^1 \\ &= (7 - \lambda \cdot 6) \cdot 5^2 \cdot 1^1 \\ &= 10 - 7\lambda. \end{aligned}$$

For the second half of columns of matrix $M(\lambda)$, we use (2). When $i = 2$, $a = 2$ and $b = 2$,

$$\begin{aligned} M(\lambda)_{2,2,2} &= -(x_2)^2 \cdot (y_2)^2 \\ &= -(3^2) \cdot (3^2) \\ &= 7. \end{aligned}$$

When $i = 2$, $a = 2$ and $b = 1$,

$$\begin{aligned} M(\lambda)_{2,2,1} &= -(x_2)^2 \cdot (y_2)^1 \\ &= -(3^2) \cdot (3^1) \\ &= 6. \end{aligned}$$

When all the entries in $M(\lambda)$ have been calculated, see Appendix A. In Appendix A, we can see that the dimension of $M(\lambda)$ is 10×18. Next, we consider $M(\lambda)$ with $\lambda = 0$ and apply Gaussian elimination to calculate the rank of matrix $M(0)$. The rank for matrix $M(0)$ is 10, which, in this example case (ii), is applied, and we take columns 9 to 18 to be a sub-square matrix of $M(\lambda)$. Then, the sub-square matrix denoted by $M'(\lambda)$ is the matrix with dimensions 10×10, as follows:

$$M'(\lambda) = \begin{bmatrix} 4 & 10 & 7 & 6 & 9 & 3 & 1 & 7 & 6 & 2 \\ 8-3\lambda & 10 & 8 & 2 & 8 & 2 & 6 & 2 & 6 & 7 \\ 6-5\lambda & 10 & 5 & 8 & 8 & 4 & 2 & 2 & 1 & 6 \\ 7-2\lambda & 10 & 9 & 7 & 7 & 3 & 6 & 6 & 1 & 2 \\ 10-7\lambda & 10 & 10 & 10 & 6 & 6 & 6 & 8 & 8 & 8 \\ 3-5\lambda & 10 & 6 & 8 & 5 & 3 & 4 & 8 & 7 & 2 \\ 8-4\lambda & 10 & 4 & 6 & 4 & 6 & 9 & 6 & 9 & 8 \\ 9 & 10 & 3 & 2 & 3 & 2 & 5 & 2 & 5 & 7 \\ 1-5\lambda & 10 & 2 & 7 & 2 & 7 & 8 & 7 & 8 & 6 \\ 5-6\lambda & 10 & 1 & 10 & 1 & 10 & 1 & 10 & 1 & 10 \end{bmatrix}.$$

Furthering the process, we calculate determinant $f(\lambda)$,

$$f(\lambda) = \det\left(M'(\lambda)\right) = 74877540\lambda - 42937040.$$

The highest degree of polynomial $f(\lambda)$ is 1. This coincides with the fact that $M'(\lambda)$ has one column that contains λ. Upon computing $f(\lambda)$ modulo $q = 11$, we obtain the following:

$$f(\lambda) = \lambda - 3.$$

We take $\lambda = 3$ as the secret key. In line with Theorem 1, $M'(3)$ is indeed a non-invertible matrix. To see this fact, we compute column matrix Y, which is the nullspace of $M'(3)$, respectively. Column matrix Y is given by

$$Y = \begin{bmatrix} 9 \\ 1 \\ 4 \\ 0 \\ 5 \\ 5 \\ 3 \\ 7 \\ 9 \\ 1 \end{bmatrix}.$$

Let us observe that $M'(3) \cdot Y = 0$.

Remark 3. *Let us assume that the adversary knows that*

$$\begin{aligned} \mu + e &= (9,8,7,5,0,10,6,5,6,9) + (0,0,0,0,0,7,0,0,0,0) \\ &= (9,8,7,5,0,6,6,5,6,9). \end{aligned} \tag{18}$$

It is easy to see that the adversary can determine whether $\lambda = 3$ is the secret key or not. This can be illustrated as follows:

$$\begin{aligned} CT - 3 \times PK &= (9,2,10,2,7,4,10,5,9,5) - 3 \times (0,9,1,10,6,3,5,0,1,6) \\ &= (9,8,7,5,0,6,6,5,6,9). \end{aligned} \tag{19}$$

Hence, $\lambda = 3$ is the correct value of α.

Remark 4. *At this point, when the adversary tries $\lambda = 3$, it does not provide any constructive information upon his attempt to successfully cryptanalyze the ciphertext. This is clear because the adversary does not have Equation (18) at hand to make a comparison with (19). The usefulness of the above strategy can only be seen in the following section, IND-CPA on the AAK-Cryptosystem.*

3.4. Indistinguishable under Chosen Plaintext Attack on AAK-Cryptosystem

This section proves that the AAK-Cryptosystem is not IND-CPA secure. Let us observe that within the AAK-Cryptosystem, the weight of small error vector e must be $w < \frac{n-k}{2}$. This means there are $n - w$ elements equal to 0 in small error vector e. Therefore, we can utilize this fact to prove that the AAK-Cryptosystem is not IND-CPA secure. The theorem for this attack is reported below.

Theorem 2. *If vector $\mu + e$ has been obtained, then the AAK-Cryptosystem is not IND-CPA secure.*

Proof of Theorem 2. The PPTA conducts the following:

1. It chooses two messages, μ_0 and μ_1, in which identical elements do not share the same position in the vector and sends it to the random oracle.
2. The random oracle relays the ciphertext, where $CT = \mu_b + \alpha \times PK + e$.
3. It computes α based on Theorem 1.
4. It computes $CT - \alpha \times PK = \mu_b + e$.
5. Since the PPTA knows about secret key α, the PPTA can check the $\mu_b + e$ vector entry positions. Due to the fact that e has vector elements equal to 0 totaling $n - w$, the PPTA can identify b.

Note that if the adversary chooses an incorrect root from $f(\lambda)$, it would result in an incorrect value of α. As such, $CT - \alpha \times PK$ would result in a meaningless vector to make a comparison with either μ_0 or μ_1. The adversary would then just choose the next root available. Since the number of roots is not exponentially large, this process is feasible.

From here, we can see that the AAK-Cryptosystem is not IND-CPA secure, because the PPTA can guess which vector μ_b is encrypted with $Pr(b' = b) = 1$. Furthermore, on a side note, the PPTA can also distinguish vector e. □

3.4.1. Algorithm for Theorem 2

The Algorithm 5 for Theorem 2 is shown below.

Algorithm 5 IND -CPA on the AAK-Cryptosystem using Theorem 2

```
Input: Messages pair $(\mu_0, \mu_1)$
Output: $b$ where $b \in \{0, 1\}$
1. PPTA chooses 2 messages, $(\mu_0, \mu_1)$ where identical elements do not share the same position in the vectors.
2. PPTA sends 2 messages to random oracle.
3. Random oracle chooses 1 message between $(\mu_0, \mu_1)$.
4. Random oracle encrypts the message and publishes $CT = \mu_b + \alpha \times PK + e$.
5. PPTA computes $\alpha$.
6. PPTA computes $CT - \alpha \times PK = \mu_b + e$.
7. PPTA check $\mu_b + e$ with $(\mu_0, \mu_1)$ to determine $b$.
```

3.4.2. Numerical Illustration of Theorem 2

This section presents a numerical illustration of IND-CPA on the AAK-Cryptosystem based on Theorem 2. Given $n = 10$, $k = 3$, $w = 1$ and $W = 3$ in $\mathbb{F}_{11}$, let $x = (2, 3, 3, 4, 5, 6, 7, 8, 9, 10)$ and $y = (4, 3, 6, 2, 1, 5, 7, 8, 9, 10)$. We take the private key,

$$p(x, y) = x^2y + xy^2 + 3xy + 5$$

and big error vector E,

$$E = (0, 0, 0, 10, 0, 7, 3, 0, 0, 0).$$

The public key is:

$$PK = C + E$$

where $C = ev(p(x, y))$; hence,

$$p(2, 4) = 0,\ p(3, 3) = 9,\ p(3, 6) = 1,\ p(4, 2) = 0,\ p(5, 1) = 6,$$

$$p(6, 5) = 7,\ p(7, 7) = 2,\ p(8, 8) = 0,\ p(9, 9) = 1,\ p(10, 10) = 6.$$

Therefore,

$$\begin{aligned} PK &= C + E \\ &= (0, 9, 1, 0, 6, 7, 2, 0, 1, 6) + (0, 0, 0, 10, 0, 7, 3, 0, 0, 0) \\ &= (0, 9, 1, 10, 6, 3, 5, 0, 1, 6). \end{aligned}$$

Two messages are chosen by the PPTA, $\mu_0(x, y) = xy + 2x + 4y + 3$ and $\mu_1(x, y) = xy + 5x + 8y + 7$. These two messages are encoded into codewords μ_0 and μ_1, respectively, where $\mu_b = ev(\mu(x, y))$ for $b \in \{0, 1\}$. For $\mu_0(x, y) = xy + 2x + 4y + 3$, it is encoded as follows:

$$\mu_0(2, 4) = 9,\ \mu_0(3, 3) = 8,\ \mu_0(3, 6) = 7,\ \mu_0(4, 2) = 5,\ \mu_0(5, 1) = 0,$$

$$\mu_0(6, 5) = 10,\ \mu_0(7, 7) = 6,\ \mu_0(8, 8) = 5,\ \mu_0(9, 9) = 6,\ \mu_0(10, 10) = 9.$$

Then, we obtain $\mu_0 = (9, 8, 7, 5, 0, 10, 6, 5, 6, 9)$. For $\mu_1(x, y) = xy + 5x + 8y + 7$, it is encoded as follows:

$$\mu_1(2, 4) = 2,\ \mu_1(3, 3) = 0,\ \mu_1(3, 6) = 0,\ \mu_1(4, 2) = 7,\ \mu_1(5, 1) = 1,$$

$$\mu_1(6,5) = 8,\ \mu_1(7,7) = 4,\ \mu_1(8,8) = 10,\ \mu_1(9,9) = 7,\ \mu_1(10,10) = 6.$$

Then, we obtain $\mu_1 = (2,0,0,7,1,8,4,10,7,6)$. The PPTA must ensure that identical elements in the two message vectors do not share the same location. Next, the PPTA sends these two message vectors, (μ_0 and μ_1) to the random oracle. The random oracle chooses one of the message vectors, encrypts it and publishes ciphertext CT, where

$$\begin{aligned} CT &= \mu_b + \alpha \times PK + e \\ &= (9,2,10,2,7,4,10,5,9,5). \end{aligned}$$

Since the value of secret key α can be computed based on Theorem 1, the PPTA retrieves $\alpha = 3$. Next, the PPTA computes equation

$$CT - \alpha \times PK = \mu_b + e$$

and obtains $\mu_b + e = (9,8,7,5,0,6,6,5,6,9)$. The PPTA can check the entry positions using Equation (19). Finally, the PPTA can identify b from $\mu_b + e$, considering the fact that e has vector elements equal to 0 totaling $n - w$. To this end, the PPTA can identify $b' = 0$ with probability equal to one.

4. Discussion

This analysis shows that from $M(\lambda)$, we choose columns 9 to 18 to be our sub-square matrix $M'(\lambda)$. In our study, we also observe that columns 7 to 16 also give the correct value of α, where the sub-square matrix is given as follows:

$$M_1'(\lambda) = \begin{bmatrix} 3 & 1 & 4 & 10 & 7 & 6 & 9 & 3 & 1 & 7 \\ 7-4\lambda & 10-\lambda & 8-3\lambda & 10 & 8 & 2 & 8 & 2 & 6 & 2 \\ 2-9\lambda & 1-10\lambda & 6-5\lambda & 10 & 5 & 8 & 8 & 4 & 2 & 2 \\ 10-6\lambda & 9-\lambda & 7-2\lambda & 10 & 9 & 7 & 7 & 3 & 6 & 6 \\ 10-7\lambda & 10-7\lambda & 10-7\lambda & 10 & 10 & 10 & 6 & 6 & 6 & 8 \\ 1-9\lambda & 5-\lambda & 3-5\lambda & 10 & 6 & 8 & 5 & 3 & 4 & 8 \\ 6-3\lambda & 9-10\lambda & 8-4\lambda & 10 & 4 & 6 & 4 & 6 & 9 & 6 \\ 1 & 8 & 9 & 10 & 3 & 2 & 3 & 2 & 5 & 2 \\ 3-4\lambda & 5-3\lambda & 1-5\lambda & 10 & 2 & 7 & 2 & 7 & 8 & 7 \\ 5-6\lambda & 6-5\lambda & 5-6\lambda & 10 & 1 & 10 & 1 & 10 & 1 & 10 \end{bmatrix}.$$

Then, the determinant for $M_1'(\lambda)$, where $f_1(\lambda) = \det(M_1'(\lambda))$, is

$$f_1(\lambda) = \det\,(M_1'(\lambda)) = -21483650\lambda^3 + 136151740\lambda^2 + 72625310\lambda + 44956500.$$

The highest degree of polynomial $f_1(\lambda)$ is 3. This coincides with the fact that $M_1'(\lambda)$ has three columns that contain λ. Upon computing $f_1(\lambda)$ modulo $q = 11$, we obtain the following:

$$f_1(\lambda) = 10(\lambda^2 + 4\lambda + 2)(\lambda - 3).$$

When $\lambda = 3$, determinants $f(\lambda)$ and $f_1(\lambda)$ are 0. From the analysis performed above, we can see that from determinant $f(\lambda)$, we can obtain a set of λ, where one of them is the correct value of α. In order to determine which root is the correct value of α, we need to compute $CT - \alpha \times PK = \mu + e$. We know that the weight of small error e must be $\frac{n-k}{2}$, which gives us the information about the zero elements in e. Next, if $\lambda \neq \alpha$, then vector $CT - \lambda \times PK$ does not provide any significant information about the message. Hence, this cryptanalysis presents a good outcome, where secret key α can be determined. Therefore, this shows that the AAK-Cryptosystem is not IND-CPA secure.

5. Conclusions

In this research study, we present an algebraic cryptanalysis of an AAK-Cryptosystem as described in [16]. This attack is resourced from strategies found in [15]. In this paper, we proved that we managed to form a list of possible values of the secret key, α. Furthering our analysis, we were able to prove that the AAK-Cryptosystem is not IND-CPA secure. As such, the AAK-Cryptosystem, as outlined in [16], is not suitable for utilization upon a set of plaintexts originating from a domain of non-exponential size.

Author Contributions: Conceptualization, S.N.Y. and M.R.K.A.; methodology, S.N.Y., M.R.K.A., T.S.C.L. and N.R.S.; validation, M.R.K.A.; formal analysis, S.N.Y.; investigation, S.N.Y., M.R.K.A., T.S.C.L. and N.R.S.; resources, M.R.K.A.; writing—original draft preparation, S.N.Y.; writing—review and editing, S.N.Y., M.R.K.A., T.S.C.L., N.R.S., S.-C.Y. and T.T.V.Y.; visualization, S.N.Y., M.R.K.A., T.S.C.L., N.R.S., S.-C.Y. and T.T.V.Y.; supervision, M.R.K.A.; project administration, M.R.K.A.; funding acquisition, T.S.C.L., S.-C.Y., T.T.V.Y. and M.R.K.A. All authors have read and agreed to the published version of the manuscript.

Funding: The research was supported by the Ministry of Higher Education Malaysia through the Fundamental Research Grant Scheme (FRGS/1/2019/STG06/UPM/02/8). It was also partially supported by the Mediterranea Universiti of Reggio Calabria (UNIRC) Research Grant (UPM/INSPEM/700-3/1/GERAN ANTARABANGSA/6380071–10065). The results of Terry Shue Chien Lau were supported by the MMU Postdoc (MMUI/220141).

Data Availability Statement: Not applicable.

Acknowledgments: The first author would like to further express appreciation to Institute for Mathematical Research (INSPEM), Universiti Putra Malaysia (UPM) and Ministry of Higher Education (MOHE) for giving the opportunity to conduct this research.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

PRP	Polynomial Reconstruction Problem
IND-CPA	Indistinguishable under Chosen Plaintext Attack
PPTA	Probabilistic Polynomial Time Adversary

Appendix A

Full matrix $M(\lambda)$ for numerical illustration of Theorem 1:

$$M(\lambda)=\begin{bmatrix}
9 & 3 & 1 & 7 & 6 & 2 & 3 & 1 & 4 & 10 & 7 & 6 & 9 & 3 & 1 & 7 & 6 & 2\\
2-9\lambda & 6-5\lambda & 7-4\lambda & 6-5\lambda & 7-4\lambda & 10-\lambda & 7-4\lambda & 10-\lambda & 8-3\lambda & 10 & 8 & 2 & 8 & 2 & 6 & 2 & 6 & 7\\
10-\lambda & 5-6\lambda & 8-3\lambda & 8-3\lambda & 4-7\lambda & 2-9\lambda & 2-9\lambda & 1-10\lambda & 6-5\lambda & 10 & 5 & 8 & 8 & 4 & 2 & 2 & 1 & 6\\
2-10\lambda & 4-9\lambda & 8-7\lambda & 8-7\lambda & 5-3\lambda & 10-6\lambda & 10-6\lambda & 9-\lambda & 7-2\lambda & 10 & 9 & 7 & 7 & 3 & 6 & 6 & 1 & 2\\
7-6\lambda & 7-6\lambda & 7-6\lambda & 2-8\lambda & 2-8\lambda & 2-8\lambda & 10-7\lambda & 10-7\lambda & 10-7\lambda & 10 & 10 & 10 & 6 & 6 & 6 & 8 & 8 & 8\\
4-3\lambda & 9-4\lambda & 1-9\lambda & 2-7\lambda & 10-2\lambda & 6-10\lambda & 1-9\lambda & 5-\lambda & 3-5\lambda & 10 & 6 & 8 & 5 & 3 & 4 & 8 & 7 & 2\\
10-5\lambda & 4-2\lambda & 6-3\lambda & 4-2\lambda & 6-3\lambda & 9-10\lambda & 6-3\lambda & 9-10\lambda & 8-4\lambda & 10 & 4 & 6 & 4 & 6 & 9 & 6 & 9 & 8\\
5 & 7 & 1 & 7 & 1 & 8 & 1 & 8 & 9 & 10 & 3 & 2 & 3 & 2 & 5 & 2 & 5 & 7\\
9-\lambda & 4-9\lambda & 3-4\lambda & 4-9\lambda & 3-4\lambda & 5-3\lambda & 3-4\lambda & 5-3\lambda & 1-5\lambda & 10 & 2 & 7 & 2 & 7 & 8 & 7 & 8 & 6\\
5-6\lambda & 6-5\lambda & 5-6\lambda & 6-5\lambda & 5-6\lambda & 6-5\lambda & 5-6\lambda & 6-5\lambda & 5-6\lambda & 10 & 1 & 10 & 1 & 10 & 1 & 10 & 1 & 10
\end{bmatrix}.$$

References

1. Brassard, G.; Lutkenhaus, N.; Mor, T.; Sanders, B.C. Security Aspects of Practical Quantum Cryptography. In Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques, Bruges, Belgium, 14–18 May 2000; pp. 289–299.
2. Cambou, B.; Gowanlock, M.; Yildiz, B.; Ghanaimiandoab, D.; Lee, K.; Nelson, S.; Philabaum, C.; Stenberg, A.; Wright, J. Post Quantum Cryptographic Keys Generated with Physical Unclonable Functions. *Appl. Sci.* **2021**, *11*, 2801. [CrossRef]

3. Shor, P.W. Algorithms for Quantum Computation: Discrete Logarithms and Factoring. In Proceedings of the 35th Annual Symposium on Foundations of Computer Science, Santa Fe, NM, USA, 20–22 November 1994; pp. 124–134.
4. Song, B.; Zhao, Y. Provably Secure Identity-Based Identification and Signature Schemes From Code Assumptions. *PLoS ONE* **2017**, *12*, e018289. [CrossRef] [PubMed]
5. Shi, J.; Chen, S.; Lu, Y.; Feng, Y.; Shi, R.; Yang, Y.; Li, J. An approach to cryptography based on continuous-variable quantum neural network. *Sci. Rep.* **2020**, *10*, 2107 . [CrossRef] [PubMed]
6. Jordan S. Quantum Algorithm Zoo. 2011. Available online: https://quantumalgorithmzoo.org/ (accessed on 5 January 2023).
7. Gaborit, P.; Otmani, A.; Kalachi, H.T. Polynomial-Time Key Recovery Attack on the Faure–Loidreau Scheme Based on Gabidulin Codes. *Des. Codes Cryptogr.* **2018**, *86*, 1391–1403. [CrossRef]
8. Imran, M.; Abideen, Z.U.; Pagliarini, S. An Experimental Study of Building Blocks of Lattice-Based NIST Post-Quantum Cryptographic Algorithms. *Electronics* **2020**, *9*, 1953. [CrossRef]
9. Naor, M.; Pinkas, B. Oblivious Transfer and Polynomial Evaluation. In Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing, Atlanta, GA, USA, 1–4 May 1999; pp. 245–254.
10. Kiayias, A.; Yung, M. Directions in Polynomial Reconstruction Based Cryptography. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* **2004**, *87*, 978–985.
11. Guruswami, V.; Sudan, M. Improved decoding of Reed-Solomon and Algebraic-Geometry Codes. *IEEE Trans. Inf. Theory* **1999**, *45*, 1757–1767. [CrossRef]
12. Augot, D.; Finiasz, M. A Public Key Encryption Scheme Based on the Polynomial Reconstruction Problem. In Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, 4–8 May 2003; pp. 229–240.
13. Kiayias, A.; Yung, M. Polynomial Reconstruction Based Cryptography. In Proceedings of the International Workshop on Selected Areas in Cryptography, Toronto, ON, Canada, 16–17 August 2001; pp. 129–133.
14. Kiayias, A.; Yung, M. Cryptanalyzing the Polynomial-Reconstruction Based Public-Key System under Optimal Parameter Choice. In Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security, Jeju, Republic of Korea, 5–9 December 2004; pp. 401–416.
15. Coron, J.S. Cryptanalysis of a Public-Key Encryption Scheme Based on the Polynomial Reconstruction Problem. In Proceedings of the International Workshop on Theory and Practice in Public Key Cryptography, Singapore, 1–4 March 2004; pp. 14–27.
16. Ajeena, R.K.; Kamarulhaili, H.; Almaliky, S.B. Bivariate Polynomials Public Key Encryption Schemes. *Int. J. Cryptol. Res.* **2013**, *4*, 73–83.
17. Lin, C.Y.; Wu, J.L. Cryptanalysis and Improvement of a Chaotic Map-Based Image Encryption System Using Both Plaintext Related Permutation and Diffusion. *Entropy* **2020**, *22*, 589. [CrossRef] [PubMed]
18. Kuwakado, H.; Morii, M. Quantum Distinguisher between the 3-Round Feistel Cipher and the Random Permutation. In Proceedings of the IEEE International Symposium on Information Theory, Austin, TX, USA, 13–18 June 2010; pp. 2682–2685.
19. Yusof, S.N.; Kamel Ariffin, M.R. An Empirical Attack on a Polynomial Reconstruction Problem Potential Cryptosystem. *Int. J. Cryptol. Res.* **2021**, *11*, 31–48.
20. Bleichenbacher, D.; Nguyen, P.Q. Noisy Polynomial Interpolation and Noisy Chinese Remaindering. In Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques, Bruges, Belgium, 14–18 May 2000; Volume 1807, pp. 53–69.
21. Sadkhan, S.B.; Ruma, K.H. Evaluation of Polynomial Reconstruction Problem using Lagrange Interpolation Method. In Proceedings of the 2006 2nd International Conference on Information and Communication Technologies, Damascus, Syria, 24–28 April 2006; Volume 1, pp. 1399–1403.
22. Augot, D.; Finiasz, M.; Loidreau, P. Using the Trace Operator to Repair the Polynomial Reconstruction Based Cryptosystem Presented at Eurocrypt 2003. *Int. Assoc. Cryptologic Res.* **2003**, 209 .
23. Zhu, S.; Han, Y. Generative Trapdoors for Public Key Cryptography Based on Automatic Entropy Optimization. *China Commun.* **2021**, *18*, 35–46. [CrossRef]
24. Carstens, T.V.; Ebrahimi, E.; Tabia, G.N.; Unruh, D. On Quantum Indistinguishability Under Chosen Plaintext Attack. *Int. Assoc. Cryptologic Res.* **2020**, 596.
25. Abdalla, M.; Benhamouda, F.; Pointcheval, D. Public-Key Encryption Indistinguishable Under Plaintext-Checkable Attacks. *IET Inf. Secur.* **2016**, *10*, 288–303. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.