Remarks on the Attack of Fouque et al. against the ℓIC Scheme

Naoki Ogura and Shigenori Uchiyama

Tokyo Metropolitan University
ogura-naoki@ed.tmu.ac.jp, uchiyama-shigenori@tmu.ac.jp

Abstract. In 2007, ℓ-Invertible Cycles (ℓIC) was proposed by Ding et al. This is one of the most efficient trapdoors for encryption/signature schemes, and of the mixed field type for multivariate quadratic public-key cryptosystems. Such schemes fit on the implementation over low cost smart cards or PDAs. In 2008, Fouque et al. proposed an efficient attack against the ℓIC signature scheme by using Gröbner basis algorithms. However, they only explicitly dealt with the odd case, i.e. ℓ is odd, but the even case; they only implemented their proposed attack in the odd case. In this paper, we propose an another practical attack against the ℓIC encryption/signature scheme. Our proposed attack does not employ Gröbner basis algorithms, and can be applied to the both even and odd cases. We show the efficiency of the attack by using some experimental results. Furthermore, the attack can be also applied to the ℓIC- scheme. To the best of our knowledge, we for the first time show some experimental results of a practical attack against the ℓIC- scheme for the even case.

1 Introduction

In 2007, Ding, Wolf and Yang [4] proposed ℓ-Invertible Cycles (ℓIC), which is a trapdoor for public-key encryption/signature schemes, and also proposed ℓIC-, which is a signature scheme as a variation of it. The security of these schemes is based on the hardness of solving simultaneous multivariate quadratic equations (MQ system) over a finite field whose characteristic is two; we call the problem "MQ problem". The problem of deciding whether an MQ system has a solution or not belongs to NP-complete, and quantum polynomial algorithms for solving the MQ problem are still unknown. It has been expected that public-key cryptosystems based on NP-complete problems like the MQ problem will replace the schemes based on the integer factorization or discrete logarithm problems which will be efficiently solved by quantum computers. Moreover, the schemes based on the MQ problem are very practical under limited computational resources, such as smart cards or PDAs. The ℓIC scheme is more efficient than Quartz[3] under the same security level.

In 2007, Dubois, Fouque, Shamir and Stern [8] proposed a practical attack against SFLASH which is called C^{*-} scheme with specific parameters. The ℓIC-signature scheme is similar to SFLASH. In 2008, Fouque, Gilles, Perret and Stern proposed an efficient attack against the ℓIC signature scheme by using Gröbner basis algorithms. However, they only considered the cryptanalysis under some condition; namely, they only considered the case that a secret-key as a pair of two transformations S, T is a pair of linear maps not affine ones. Also, they only explicitly dealt with the odd case, i.e. ℓ is odd, but the even case; they only implemented their proposed attack in the odd case.

In this paper, we propose an another practical attack against the ℓIC encryption/signature scheme. Our proposed attack does not employ Gröbner basis algorithms, and can be applied to the both even and odd cases. We show the efficiency of the attack by using some experimental results under the most general condition that a secret-key as a pair of two transformations S, T is a pair of affine ones. To the best of our knowledge, we for the first time show some experimental results of a practical attack against the ℓIC- signature scheme for the even case.

This paper is organized as follows. In Section 2, we will briefly introduce ℓ-Invertible Cycles. In Section 3, 4, we will propose the attack against ℓIC. In Section 5, we will show our experimental results. In Section 6, we will conclude this paper.

2 ℓ-Invertible Cycles

In this section, we will describe about public-key cryptosystems based on the multivariate quadratic equations (Multivariate Quadratic public-key Cryptography: MQPKC), and ℓIC (ℓ-Invertible Cycles) and ℓIC- schemes.

2.1 MQPKC

We will briefly review about public-key cryptosystems based on the MQ problem. Matsumoto and Imai[16] proposed the first MQPKC scheme. In the last few decades, there has been enormous amount of work devoted to this area. Most public-key cryptosystems based on the MQ problem use a trapdoor function below. For two vector spaces K^{n_1} and K^{n_2} over a finite field $K = \mathbb{F}_q$ whose characteristic is 2, we construct a public-key as a map $P : K^{n_1} \to K^{n_2}$. Each scheme has the map $F : K^{n_1} \to K^{n_2}$ whose inverse images are efficiently computable. This map, which is constructed by the n_2 quadratic polynomials with n_1 variables, is called "central map." We generate a secret-key as a pair of two bijective affine transformations (S, T). We call these transformations "secret transformations." Of course, the inverse images of secret transformations is easy to compute if these transformations are known. Finally, the map $P = T \circ F \circ S$ is made to public. Since it is believed that solving the MQ problem is intractable, we may assume that it is difficult to compute an inverse image for the public-key if secret transformations are unknown.

For the encryption scheme use, Alice (a sender) sends a ciphertext $C = P(m)$, where m is a plaintext and P is Bob's public-key. Bob (a receiver) obtains $m = P^{-1}(C)$, where $P^{-1} = S^{-1} \circ F^{-1} \circ T^{-1}$. For the signature scheme use, on the other hand, Alice (a signer) generates a signature $\sigma = P^{-1}(m)$, where m is a message and P is Alice's public-key. Bob (a verifier) checks whether $m = P(\sigma)$ or not.

Many central maps are classed as four basic types below.

- MIA (Matsumoto-Imai Scheme A, C^*)[16], [18]
- HFE (Hidden Field Equations)[21]
- UOV (Unbalanced Oil and Vinegar)[20], [17]
- STS (Stepwise Triangular Systems)[15], [25]

Although these schemes are efficient, some of them are broken. So some modifiers have been proposed for enforcing schemes against known attacks. For more information about the modifiers, [24] is very useful for us. Ding et al. [4] recommended three modifiers for ℓIC schemes below.

- Minus[23]
- Internal Perturbed Plus[21], [7], [6], [5]
- Embedding[4], similar to Fixing[2]

We will describe ℓIC-, which is a variation of ℓIC with the Minus method later.

2.2 ℓIC

We will show the central map based on ℓIC scheme[4]. Let $\ell \geq 2$ be a divisor of $n := n_1 = n_2$. Define L by an n/ℓ-dimensional extended field of $K = \mathbb{F}_q$. An L-vector space L^ℓ can be identified to a K-vector space K^n with a fixed basis over K. Fix the values $(\lambda_1, \ldots, \lambda_\ell) \in \{0, \ldots, n/\ell - 1\}^\ell$. Then, the central map of ℓIC scheme is

$$F : L^\ell \to L^\ell;\ (A_1, \ldots, A_\ell) \mapsto ({A_1}^{q^{\lambda_1}} A_2, \ldots, {A_\ell}^{q^{\lambda_\ell}} A_1)\ . \tag{1}$$

In addition, in [4], the concrete values of λ_i were proposed due to efficiency of computation of the inverse image:

$$\lambda_1 = \begin{cases} 0\ (\ell : \text{odd}) \\ 1\ (\ell : \text{even}) \end{cases},\ \lambda_i = 0\ (2 \leq i \leq \ell) \tag{2}$$

In the case when ℓ is even, the condition that $q = 2$ (i.e. $K = \mathbb{F}_2$), are required for bijectivity of the central map. Then, for $A = (A_1, \ldots, A_\ell)$, the explicit central map of these schemes is given as follows.

$$F(A) = \begin{cases} (A_1A_2,\ A_2A_3, \ldots, A_\ell A_1)\ (q: \text{ a power of two})\ \text{for } \ell : \text{odd} \\ (A_1^2A_2,\ A_2A_3, \ldots, A_\ell A_1)\qquad (q = 2)\qquad \text{for } \ell : \text{even} \end{cases} \tag{3}$$

We call these scheme odd-IC scheme and even-IC scheme, respectively.

2.3 ℓIC-

In [4], some schemes with modifiers were also proposed due to enforcing an ℓIC scheme. The ℓIC- scheme is the scheme applied the Minus method to the ℓIC scheme. Select an integer $r < n$, and define a projection map $\Pi : K^n \to K^{n-r}$. This scheme uses not P but $\Pi P = \Pi \circ P$ as a public-key. This modification prevents attackers from breaking the scheme by using bijectivity of P. Because of filling the requirements for security, r has to be some big integer. So this scheme is used not an encryption but a signature scheme.

Ding et al.[4] recommended a signature scheme based on a 3IC- scheme. But they did not mention about specific construction of the scheme. We will describe the possible concrete construction of this scheme below. This construction is the same as that of SFLASH[1].

[Secret-key]

1. Generate the bijective affine transformation $S : K^n \to K^n$. To be more precise, generate randomly an n-dimensional non-singular matrix S_L and an n-dimensional vector S_C over K, and set a function $S(x) = S_L x + S_C$.
2. Similarly, generate the affine transformation $T : K^n \to K^n$ (i.e. a matrix T_L and a vector T_C).
3. Generate a random binary string Δ (enough long to satisfy security requirements).

Let a secret-key be $(S,\ T,\ \Delta)$.

[Public-key]
Construct a map $P = T \circ F \circ S$, where F is the bilinear map over $(L^*)^\ell$ below.

$$F((A_1,\ A_2,\ A_3)) = (A_1 A_2,\ A_2 A_3,\ A_3 A_1)\ (A_i \in L) \tag{4}$$

Recall that K^n is identified to L^ℓ. By a projection $\Pi : K^n \to K^{n-r}$, let a public-key be $\Pi \circ P$.

[Signing]
We regard the binary strings with length $\lg q$ as the elements over K.

1. By applying a hash function to a message, generate $V \in K^{n-r}$.
2. By applying a hash function to the concatenation between V and Δ, generate $W \in K^r$.
3. By concatenating between V and W, generate $B \in K^n$. Let a signature be $P^{-1}(B) = S^{-1}(F^{-1}(T^{-1}(B)))$.

[Verification]

1. By applying the hash function to the message, generate $Y \in K^{n-r}$.
2. Verify that Y corresponds with the element generated by applying $\Pi \circ P$ to the signature.

3 Attack against ℓIC

In this section, we will give an attack against ℓIC schemes.

3.1 Summary of Attack against ℓIC

In this subsection, we will show a brief sketch of our attack against ℓIC. The Table 1 shows an algorithm of breaking both a signature scheme (forging) and an encryption scheme (deciphering) based on odd-IC schemes. The attack will be applied to other ℓIC schemes. The attack against ℓIC consists of 2 steps below.

- Reduction to linear transformation
- Forging a signature(resp. Deciphering)

In this attack, we employ a property of the easily invertible central map F, which is taken over a public-key P. The MQ problem is reduced to the linear system problem.

Table 1. The algorithm of an attack against odd-IC scheme

Input: a public-key P, parameters q, n, ℓ, a message m (resp. a ciphertext c)
Output: the valid signature against m (resp. the plaintext for c)
{**(Reduction to linear transformation)**}
$\{\nu_j\} \leftarrow$ Linear terms of P
$\{\zeta_{ij}\} \leftarrow$ Quadratic (not double) terms of P
$v$space $\leftarrow$ Kernel of $\{\nu_j\}$
$v \leftarrow$ A vector of the $v$space satisfying equation(6)
$P_L(x) := P(x - v) - P(-v)$
{**(Forging (resp. Deciphering))**}
$\{(\gamma^{(k)},\ \gamma'^{(k)})\} \leftarrow \{(\gamma^{(k)},\ \gamma'^{(k)})\}$ satisfying forging equations (11)
$V \leftarrow$ The element over K^n generated by applying hash function to m (resp. c)
$y \leftarrow V - P(-v)$
$x$space $\leftarrow$ Space of x satisfying forging equations (11) for y
$x \leftarrow$ A vector of the $x$space satisfying $y = P_L(x)$
return $x - v$

3.2 Reduction to Linear Transformation

Some attacks against MQPKC need that S and T are not only the affine but the linear transformations. The reduction attacks in the cases of SFLASH [14], [13] and HFE [10] were proposed. This process is used for the attack by using a differential against the Minus method. Moreover, this reduction is also useful for our attack against ℓIC schemes. We will see in the next section.

As is well known, a polynomial is called a quadratic form or a homogeneous polynomial if all terms of it have degree 2. As for ℓIC, each coordinate of F is a quadratic form. We will see how to use this property. The aim of this attack is to find $v = (v_1, \ldots, v_n) := S_L^{-1} \cdot S_c$. Since each coordinate of F is a quadratic form, that of $P_L(x) := P(x - v) - P(-v)$ is also the quadratic form, whose coordinate consists of only quadratic (including double) terms of that of P. Here, define the differential[12] by a bilinear function below. This function is useful for cryptanalysis against the schemes based on the MQ problem.

$$DP(a,\ b) := P(a + b) - P(a) - P(b) + P(0) \tag{5}$$

We will describe in detail the way of using the differential for an attack against the Minus method later. By definition of DP, the equation below can be easily shown.

$$-DP(x,\ -v) = P(x) - P_L(x) - P(0) \tag{6}$$

Because the right hand of (6) consists of only linear terms of P, it is efficiently computable from the public-key P under the condition that $K \neq \mathbb{F}_2$. Then, we can compute v by solving the linear equations created from (6). Since $(a,\ P(a))$ corresponds to $(b = a + v,\ P_L(b) = P(a) - P(-v))$, the function P_L, whose S and T are linear, give information about the public-key P.

3.3 Forging Equation

We will adapt the attack against C^* scheme[22] to ℓIC scheme. Because of concise representation of index, let μ be the function below for each $i \in \mathbb{Z}$. Note that this symbol is different from [4]'s one.

$$\mu(i) := \begin{cases} \ell & (\ell \mid i) \\ i \mod \ell & (\text{otherwise}) \end{cases} , \tag{7}$$

where mod express the least non-negative residue. This residue is called the "least positive residue".

The obvious relation below is derived between $(B_1, \ldots, B_\ell) = F((A_1, \ldots, A_\ell)) = ({A_1}^{q^{\lambda_1}} A_2, \ldots, {A_\ell}^{q^{\lambda_\ell}} A_1)$.

$$A_{\mu(i+2)} {B_i}^{q^{\lambda_{\mu(i+1)}}} - {A_i}^{q^{\lambda_i + \lambda_{\mu(i+1)}}} B_{\mu(i+1)} = 0 \tag{8}$$

Let y be $P(x)$. Then, $T^{-1}(y) = F(S(x))$, where S and T are some bijective affine transformation. By applying (8) to this equation, the simple relation below is found.

$$\sum_{1 \leq i,j \leq n} {\gamma_{ij}}^{(k)} x_i y_j + \sum_{i=1}^{n} ({\alpha_i}^{(k)} x_i + {\beta_i}^{(k)} y_i) + \delta^{(k)} = 0 , \tag{9}$$

where ${\gamma_{ij}}^{(k)},\ {\alpha_i}^{(k)},\ {\beta_i}^{(k)},\ \delta^{(k)} \in K$. This relation induces linear equations between x and $y(= P(x))$. We call this relation **forging equation**.

Here, let us examine the number of linear independent equations, i.e. the rank of the coefficient matrix. In the case of even-IC scheme, almost all y makes full rank system. So we can easily forge signatures based on even-IC schemes by using equations (9). In the case of odd-IC scheme, On the other hand, unfortunately an linear dependent equation exists because $\lambda_1 = \lambda_2 = \cdots = \lambda_\ell = 0$. Thus, (9) gives us only $(n-k)$-dimensional equations, so we need exhaustive search from k-dimensional space.

Odd Case In this subsection, we will give different relation from (8). Put $(B_1, \ldots, B_\ell) = F((A_1, \ldots, A_\ell)) = (A_1 A_2, \ldots, A_\ell A_1)$ for an odd-IC scheme. Using the technique of computation of inverse images shown in [4], we can obtain the relation below.

$$\prod_{j=0}^{(\ell-1)/2} B_{\mu(i+2j)} = {A_i}^2 \prod_{j=0}^{(\ell-1)/2-1} B_{\mu(i+2j+1)} \tag{10}$$

Note that the map $A_i \mapsto {A_i}^2$ is bijective since A_i is in K, whose characteristic is two. Like the previous subsubsection, for $y = P(x)$, we can find the relation $q_1(y_1, \ldots, y_n) = q_2(x_1, \ldots, x_n, y_1, \ldots, y_n)$, where the total degree of the polynomial q_1 and q_2 is $(\ell+1)/2$.

Now we show an advantage of linearity of S and T. By applying (10) to $y = P(x)$ in this case, more simple relation below is found.

$$\sum_{|\boldsymbol{w_1}|=(\ell+1)/2} {\gamma_{\boldsymbol{w_1}}}^{(k)} y_{\boldsymbol{w_1}} = \sum_{i=1}^{n} \sum_{|\boldsymbol{w_2}|=(\ell-1)/2} {\gamma'_{i\boldsymbol{w_2}}}^{(k)} {x_i}^2 y_{\boldsymbol{w_2}} , \tag{11}$$

where we define $y_{\boldsymbol{w}} = y_{c_1} \cdots y_{c_{|\boldsymbol{w}|}}$ for $y = (y_1, \ldots, y_n) \in K^n$ and $\boldsymbol{w} = (c_1, \ldots, c_{|\boldsymbol{w}|})$ such that $1 \leq c_i \leq n$. For example, in the case that $\ell = 3$, the equation below is induced.

$$\sum_{1 \leq i_1 \leq i_2 \leq n} {\gamma_{i_1 i_2}}^{(k)} y_{i_1} y_{i_2} = \sum_{1 \leq i,j \leq n} {\gamma'_{ij}}^{(k)} {x_i}^2 y_j \tag{12}$$

Thus the reduction attack in Section 3.2 has the advantage that we can save computational space.

4 Attack against ℓIC-

We will describe an attack against ℓIC-, which is a Minus variation of ℓIC.

4.1 Summary of Attack against ℓIC-

In this section, we will show a brief sketch of our attack against ℓIC- schemes. The Table 2 shows an algorithm of breaking signature schemes based on odd-IC$^-$ scheme. The attack against ℓIC- consists of 3 steps below.

- Reduction to linear transformation
- Recovering the deleted part by using a differential
- Forging a signature

This attack is summarized as follows. First of all, the secret affine transformations S and T are reduced to the linear transformation. Secondly, we recover the part which is deleted by the projection in the time of transforming ℓIC to ℓIC-. Finally, we can apply the attack against ℓIC.

The first and third step were already showed at the previous section. In the next subsection, we will explain details of the second step.

Table 2. The algorithm of an attack against odd-IC$^-$ scheme

Input: a public function ΠP, parameters $q,\ n,\ \ell,\ r$, a message m
Output: the valid signature for m
$\{$**(Reduction to linear transformation)**$\}$
$\{\nu_j\} \leftarrow$ Linear terms of ΠP
$\{\zeta_{ij}\} \leftarrow$ Quadratic (not double) terms of ΠP
$v$space $\leftarrow$ Kernel of $\{\nu_j\}$
$v \leftarrow$ A vector of the $v$space satisfying equation(6)
$\Pi P_L(x) := \Pi P(x - v) - \Pi P(-v)$
$\{$**(Recovering the deleted part by using a differential)**$\}$
$\Pi DP_L(a,\ x) := \Pi P_L(x + a) - \Pi P_L(x) - \Pi P_L(a)$
Nspace $\leftarrow$ Space of N which is kernel of function (21)
while true do
$\{$**(Forging)**$\}$
$N_\xi \leftarrow$ Regular(not scalar) matrix of Nspace
$P_f \leftarrow$ Function $K_n \rightarrow K_n$(full rank) by adding $(\Pi P_L) \circ N_\xi$ to ΠP_L
γspace $\leftarrow \{(\gamma^{(k)},\ \gamma'^{(k)})\}$ satisfying forging equations (11)
if rank(γspace) $\leq n$ then
$\{(\gamma^{(k)},\ \gamma'^{(k)})\} \leftarrow \gamma$space
break
end if
end while
$V \leftarrow$ The element over K^{n-r} generated by applying hash function to m
$y \leftarrow$ The element over K^n generated by $V - \Pi P(-v)$ (with random padding)
$x$space $\leftarrow$ Space of x satisfying forging equations (11) for y
$x \leftarrow$ A vector of the $x$space satisfying $y = P_L(x)$
return $x - v$

4.2 Recovering with Differential

In [11], ℓIC- schemes were broken. This is an application of the attack against SFLASH[8], [9]. We will describe in detail this attack. In what follows, we assume that secret transformations are the linear transformation. Note that the reduction attack is not applied to even-IC schemes because K is a prime field $\mathbb{F}_2$.

For $\xi := (\xi_1, \ldots, \xi_\ell) \in L^\ell$, define

$$\begin{aligned} M_\xi : L^\ell &\to L^\ell \\ (A_1, \ldots, A_\ell) &\mapsto (\xi_1 A_1, \ldots, \xi_\ell A_\ell) \ . \end{aligned} \tag{13}$$

We can check $F \circ M_\xi = M_{F(\xi)} \circ F$. Here, remember that a differential $DF(A,\ B) = F(A+B) - F(A) - F(B) + F(0)$. For $A = (A_1, \ldots, A_\ell)$ and $B = (B_1, \ldots, B_\ell)$, we have the specific form of DF below.

$$DF(A,\ B) = ({A_1}^{q^{\lambda_1}} B_2 + {B_1}^{q^{\lambda_1}} A_2, \ldots, {A_\ell}^{q^{\lambda_\ell}} B_1 + {B_\ell}^{q^{\lambda_\ell}} A_1) \tag{14}$$

So we can see

$$DF(M_\xi(A),\ B) + DF(A,\ M_\xi(B)) = L_\xi(DF(A,\ B)) \ , \tag{15}$$

where $L_\xi := M_{({\xi_1}^{q^{\lambda_1}} + \xi_2, \ldots, {\xi_\ell}^{q^{\lambda_\ell}} + \xi_1)}$.

A differential has various properties. If S is linear, the relation between DF and DP is easily shown as follows.

$$DP(a,\ b) = (T \circ DF)(S(a),\ S(b)) \tag{16}$$

Thus, DP, which is generated by a public-key P, takes over properties of DF. Let N_ξ be $S^{-1} \circ M_\xi \circ S$, let N'_ξ be $T \circ L_\xi \circ T^{-1}$. The property (16) gives us the useful equation below.

$$\begin{aligned} (\Pi \circ DP)(N_\xi(a),\ b) + (\Pi \circ DP)(a,\ N_\xi(b)) \\ = \Pi(N'_\xi(DP(a,\ b))) \end{aligned} \tag{17}$$

Let BS_{N_ξ} be the function which represents left side of (17). Then, we obtain the simple relation below.

$$BS_{N_\xi}(a,\ b) = \Pi(N'_\xi(DP(a,\ b))) \tag{18}$$

We will see how to compute N_ξ by using this relation in the next subsubsection.

If once we find N_ξ, we can recover the projected part. By the definition of N_ξ, we can arrive at the useful property as the following.

$$\begin{aligned} (\Pi \circ P)(N_\xi(x)) &= \Pi(T(F(M_\xi \circ S)(x))) \\ &= \Pi((T \circ M_{F(\xi)})(F \circ S)(x)) \end{aligned} \tag{19}$$

Then recovering the deleted part by the projection Π is done as stated below.

$$P_f := (P^{(1)}, \ldots,\ P^{(n-r)},\ P^{(1)} \circ N_\xi, \ldots,\ P^{(r)} \circ N_\xi) \tag{20}$$

Note that P_f does not have to correspond with original P because a verifier checks $\Pi(P_f)$ only.

Compute N_ξ on Odd Case We consider a way of computation of N_ξ. To find a solution of N_ξ, we apply the technique of [8] or [9]. The difference of the attack between [8] and [9] is caused by whether kernel of L_ξ is trivial or not. On the context of odd-IC scheme, by the definition that $L_\xi := M_{(\xi_1+\xi_2,\ldots,\xi_\ell+\xi_1)}$, we can find a non-trivial kernel, which is $\xi_1 = \xi_2 = \cdots = \xi_\ell$. In other words, a solution that the right side of (18) equals 0 exists. This leads us a computation method of N_ξ, i.e. solving kernel of the function below.

$$(\Pi \circ DP)(N_\xi(a),\ b) + (\Pi \circ DP)(a,\ N_\xi(b)) \tag{21}$$

By the cryptanalysis similar to that of [9], we expect that N_ξ is computable under the condition that $r \leq n-3$.

Compute N_ξ on Even Case On the other hand, in the case of even-IC scheme, we can find only a trivial kernel. This is because $L_\xi = M_{({\xi_1}^2+\xi_2,\ldots,\xi_\ell+\xi_1)} = 0$ is equivalent to $\xi_1 = \xi_2 = \cdots = \xi_\ell \in \mathbb{F}_2$. So we apply the technique of [8]. Let c_{ij} be the $(i,\ j)$-element of N'_ξ. Then the equation (18) is equivalent to the relation below for $i = 1, \ldots, n-r$.

$$BS^{(i)}_{N_\xi} = \sum_{j=1}^{n} c_{ij} DP^{(j)} \tag{22}$$

We cannot solve directly the equation (22) since $DP^{(n-r+1)}, \ldots, DP^{(n)}$ are included in it. However, we expect that, for some i, $BS^{(i)}_{N_\xi}$ is in the space generated by only $DP^{(j)}$ $(j = 1, \ldots, \underline{n-r})$. We can solve this relations for $i = 1, 2, 3$ and obtain non-trivial N_ξ. Applying the [8]'s analysis to this scheme, N_ξ can be probably induced under the condition that r is up to $(n-2)/3$. [1] Our experimental results implies that this assertion is almost correct.

5 Experimental Results

In this section, we will show some computational results of attacks against ℓIC and ℓIC-. The computational complexity of our attack will be discussed.

5.1 ℓIC Case

The following is a summary of our attack against ℓIC schemes.

1 Reduction to linear transformation
2 Forging a signature (resp. Deciphering)
  2–1 Finding forging equations (9), (11)
  2–2 Forging a signature (resp. Deciphering) by using the forging equations

[1] In [8], the idea of breaking SFLASH under the case that $r < n/2$ was also discussed.

Note that we start this algorithm directly from step 2 in the case that $K = \mathbb{F}_2$.

Main computation of this algorithm against odd-IC scheme is solving $O(n^{(\ell+1)/2})$-dimensional linear equations over $\mathbb{F}_q$ in step2–1. So the asymptotic computational complexity of this algorithm is

$$O(n^{3(\ell+1)/2} \lg^2 q) \ . \tag{23}$$

Similarly, the complexity against even-IC scheme is $O(n^6 \lg^2 q)$.

We used the computer whose CPU is 2GHz AMD Opteron 246, memory is 4GB, and hard disk is 160GB. Magma[26] was used as a software for writing the program. The Table 3,4 shows experimental results of our attack.

Comparing our attack with [11], our attack would not be so efficient. This is because our attack is simple and completely analogous of Patarin's attack [22]. Fouque et al.[11] stated that Patarin's attack is not efficient and used a technique of Gröbner basis. However, the results show that Patarin's attack is still practical and polynomial time under the condition that ℓ is small.

Table 3. Experimental Results against oddIC (over $q = 2^8$)

ℓ	3	3	5
n	69	138	35
k	23	46	7
time[s]	1353.1	73814.7	82309.1

Table 4. Experimental Results against evenIC (over $q = 2$)

ℓ	2	2	4	6
n	120	240	240	240
k	60	120	60	40
time[s]	327.1	5630.3	5618.9	5668.3

5.2 ℓIC- Case

We will explain about our attack experiments against ℓIC- schemes. In this section, we classify our algorithm into 5 steps below.

1 Reduction to linear transformation
2 Recovering the deleted part by using a differential
  2–1 Finding linear equations using a differential
  2–2 Solving the equations and recovering the part deleted by a projection

3 Forging a signature
  3–1 Finding forging equations (9), (11)
  3–2 Forging a signature by using the forging equations

This method does not ensure whether we obtain non-trivial $N_\xi = S^{-1} \circ M_\xi \circ S$ in step 2–2. So our algorithm checks whether we can find non-trivial forging equations (9) in step 3–1. Computational results show that almost all parameters pass this check.

Once forging equation(9) are found, we can forge a signature by only executing step 3–2, which is an easy task. The computational complexity of this algorithm is the same as that of the ℓIC case above. The main operations of our algorithm are step 2 and step 3–1.

Results of attack experiments against proposed parameter at [4] is shown at the Table 5. We also show some experimental results against 2IC- schemes under the condition that secret transformations are linear at the Table 6.

Table 5. Experimental Results against 3IC- (over $q = 2^8$)

n	30	36	48
k	10	12	16
r	10	12	16
time[s]	34.1	79.3	321.6

Table 6. Experimental Results against 2IC- with linear (over $q = 2$)

n	40	60	80
k	20	30	40
r	10	15	20
time[s]	317.3	2021.3	7725.6

6 Conclusion

We proposed a practical attack against the ℓIC schemes. This attack can be applied to the ℓIC- signature scheme under the most general condition that secret transformations are linear. Also, oddIC$^-$ schemes (with affine secret transformations) for small ℓ are efficiently broken by our proposed attack. Furthermore, we carried out computational experiments of the attack against 3IC- signature scheme with the recommended parameters in [4] and showed that forging a signature is efficiently computable. Here we note that these attacks wold not be applied directly to solving the all MQ problems. Main emphasis of our attack is that, by using a property of the central map F, the MQ problem is reduced to solving the linear equations. We used the property to obtain forging equations, which is linear relations between the domain and the image of a public function P. Also, the differential DF enables us to fill the deleted part by a projection. Fouque et al.[11] stated that Patarin's attack would not be efficient, so they used Gröbner basis algorithms. On the other hand, our experimental results explicitly showed that Patarin's attack is still practical and polynomial time under the condition that ℓ is small.

References

1. N. Courtois, L. Goubin, and J. Patarin. "SFLASH$^{\mathrm{v3}}$, a fast asymmetric signature scheme." *Cryptology ePrint*, `http://eprint.iacr.org/2003/211` (2003)
2. N. T. Courtois. "The security of Hidden Field Equations (HFE)." *The Cryptographer's Track at RSA*, LNCS 2020, pp. 266-281, Springer (2001)
3. N. Courtois, L. Goubin, and J. Patarin. "Quartz, 128-bit long digital signatures." *The Cryptographer's Track at RSA*, LNCS 2020, pp. 298-307, Springer (2001)
4. J. Ding, C. Wolf, and B. Yang. "ℓ-Invertible Cycles for Multivariate Quadratic(MQ) Public Key Cryptography." *PKC 2007*, LNCS 4450, pp. 266-281, Springer (2007)
5. J. Ding, and J. Gower. "Inoculating multivariate schemes against differential attacks." *PKC 2006*, LNCS 3958, pp. 290-301, Springer (2006)
6. J. Ding, and D. Schmidt. "Cryptanalysis of HFEv and internal perturbation of HFE." *PKC 2005*, LNCS 3386, pp. 288-301, Springer (2005)
7. J. Ding. "A new variant of the Matsumoto-Imai cryptosystem through perturbation." *PKC 2004*, LNCS 2947, pp. 305-318, Springer (2004)
8. V. Dubois, P. A. Fouque, A. Shamir, and J. Stern. "Practical Cryptanalysis of SFLASH." *Cryptology ePrint*, `http://eprint.iacr.org/2007/141` (2007)
9. V. Dubois, P. A. Fouque, and J. Stern. "Cryptanalysis of SFLASH with Slightly Modified Parameters." *EUROCRYPT '07*, LNCS 4515, pp. 264-275, Springer (2007)
10. P. Felke. "On the Affine Transformations of HFE-Cryptosystems and Systems with Branches." *Coding and Cryptography*, LNCS 3969, pp. 229-241, Springer (2005)
11. P. A. Fouque, Gilles M. R, L. Perret, J. Stern. "Total Break of the ℓ-IC Signature Scheme." *PKC 2008*, LNCS 4939, pp. 1-17, Springer (2008)
12. P. A. Fouque, L. Granboulan, and J. Stern. "Differential Cryptanalysis for Multivariate Schemes." *EUROCRYPT '05*, LNCS 3494, pp. 341-353, Springer (2005)
13. W. Geiselmann, and R. Steinwandt. "A short comment on the affine parts of SFLASH^{v3}." *Cryptology ePrint*, `http://eprint.iacr.org/2003/220` (2003)
14. W. Geiselmann, R. Steinwandt, and Th. Beth. "Attacking the Affine Parts of SFLASH." *Cryptography and Coding*, pp. 355-359, Springer (2001)
15. L. Goubin, and N. T. Courtois. "Cryptanalysis of the TTM cryptosystem." *ASIACRYPT '00*, LNCS 1976, pp. 44-57, Springer (2000)
16. H. Imai, and T. Matsumoto. "Algebraic methods for constructing asymmetric cryptosystems." *Algebraic Algorithms and Error-Correcting Codes*, LNCS 299, pp. 108-119, Springer (1985)
17. A. Kipnis, J. Patarin, and L. Goubin. "Unbalanced Oil and Vinegar signature schemes." *EUROCRYPT '99*, LNCS 1592, pp. 206-222, Springer (1999)

18. T. Matsumoto, and H. Imai. “Public Quadratic Polynomial-tuples for Efficient Signature-Verification and Message-Encryption.” *EUROCRYPT '88*, LNCS 330, pp. 288-298, Springer (1988)
19. J. Patarin, N. Courtois, and L. Goubin. “FLASH, a Fast Multivariate Signature Algorithm.” *The Cryptographer's Track at RSA*, LNCS 2020, pp. 298-307, Springer (2001)
20. J. Patarin. “The oil and vinegar signature scheme.” *Dagstuhl Workshop on Cryptography*, transparencies (1997)
21. J. Patarin. “Hidden Field Equations (HFE) and Isomorphisms of Polynomials (IP): two new families of asymmetric algorithms.” *EUROCRYPT '96*, LNCS 1070, pp. 33-48, Springer (1996)
22. J. Patarin. “Cryptanalysis of the Matsumoto and Imai Public Key Scheme of Eurocrypt'88.” *CRYPTO '95*, LNCS 963, pp. 248-261, Springer (1995)
23. A. Shamir. “Efficient signature schemes based on birational permutations.” *CRYPTO '93*, LNCS 773, Springer (1993)
24. C. Wolf. “Multivariate Quadratic Polynomials in Public Key Cryptography.” Ph.D. Thesis, `http://hdl.handle.net/1979/148` (2005)
25. C. Wolf, A. Braeken, and B. Preneel. “Efficient cryptanalysis of RSE(2)PKC and RSSE(2)PKC.” *Security in Communication Networks*, LNCS 3352, pp. 294-309, Springer (2004)
26. Magma, `http://magma.maths.usyd.edu.au/magma/`