Cryptanalysis of GGH Map[⋆]

Yupu Hu and Huiwen Jia

ISN Laboratory, Xidian University, 710071 Xi'an, China
yphu@mail.xidian.edu.cn chusheng4@sina.com

Abstract. Multilinear map is a novel primitive which has many cryptographic applications, and GGH map is a major candidate of multilinear maps. GGH map has two classes of applications, which are respectively applications for public tools of encoding and hidden tools of encoding. In this paper we show that applications of GGH map for public tools of encoding are not secure. We present an efficient attack on GGH map, aiming at multi-party key exchange (MPKE) and the instance of witness encryption (WE) based on the hardness of 3-exact cover problem. First, for the secret of each user, we obtain an equivalent secret, which is the sum of original secret and a noise. The noise is an element of the specific principal ideal, but its size is not small. To do so, we use weak-DL attack presented by authors of GGH map. Second, we use special modular operations, which we call modified encoding/decoding, to filter the decoded noise much smaller. Such filtering is enough to break MPKE. Moreover, such filtering negates K-GMDDH assumption, which is the security basis of an ABE. The procedure almost breaks away from those lattice attacks and looks like an ordinary algebra. The key point is our special tools for modular operations. Finally, we break the instance of WE based on the hardness of 3-exact cover problem. To do so, we not only use modified encoding/decoding, but also (1) introduce and solve "combined 3-exact cover problem", which is a problem never hard to be solved; and (2) compute Hermite normal form of the specific principal ideal. The attack on the instance of WE is under an assumption, which seems at least nonnegligible.

Keywords: Multilinear maps, Multi-party key exchange (MPKE), Witness encryption (WE), Lattice based cryptography.

1 Introduction

1.1 Background and Our Contributions

Multilinear map is a novel primitive. It is the solution of a long-standing open problem [1], and has many novel cryptographic applications, such as multi-party key exchange (MPKE) [2], witness encryption (WE) [3–9], obfuscation [7–10], and so on. It also has several advantages in traditional cryptographic area [3],

[⋆] The work was supported in part by Natural Science Foundation of China under Grant 60833008

such as IBE, ABE [11], broadcasting Encryption, and so on. The first candidate of multilinear map is GGH map [2], and GGHLite map [12] is a special version of GGH map, for the purpose of improving the efficiency. Up to now, GGH map is a major candidate of multilinear maps. It uses noised encoding to obtain the trapdoor, and its security was seemly based on the hardness of several problems over lattices. GGH map has two classes of applications. The first class is applications for public tools of encoding, for example, MPKE [2] and the instance of WE based on the hardness of 3-exact cover problems [3]. The second class is applications for hidden tools of encoding, for example, GGHRSW obfuscation [7]. Authors [2] provided a survey of relevant cryptanalysis techniques from the literature, and also provided two new attacks on GGH map, as a reminding. We emphasize that they presented weak-DL attack which is primary version of our attack, and which did not find major danger.

In this paper we show that applications of GGH map for public tools of encoding are not secure. We present an efficient attack on GGH map, aiming at MPKE and the instance of WE based on the hardness of 3-exact cover problem. First, for the secret of each user, we obtain an equivalent secret, which is the sum of original secret and a noise. The noise is an element of the specific principal ideal, but its size is not small. To do so, we use weak-DL attack [2]. Second, we use special modular operations, which we call modified encoding/decoding, to filter the decoded noise much smaller. Such filtering is enough to break MPKE. Moreover, such filtering negates K-GMDDH assumption (Assumption 5.1 of [11]), which is the security basis of the ABE [11]. The procedure almost breaks away from those lattice attacks and looks like an ordinary algebra. The key point is our special tools for modular operations. Finally, we break the instance of WE based on the hardness of 3-exact cover problem. To do so, we not only use modified encoding/decoding, but also (1) introduce and solve "combined 3-exact cover problem", which is a problem never hard to be solved; and (2) compute Hermite normal form of the specific principal ideal. The attack on the instance of WE is under an assumption, which seems at least nonnegligible.

1.2 Principles and Meanings of Our Attack

Quite different from original DH maps and bilinear maps, all candidates of multilinear maps have a common security worry that decoding tools are public. This makes the adversary decode messages freely. The adversary can choose those decoded messages that are small enough, without protection of the modular operation. Such security worry has been used to break CLT map [13–17], which is another candidate of multilinear map, and which is simply over integers. Multilinear maps over the integer polynomials (GGH map [2] and GGHLite map [12]) haven't been broken because (1) (NTRU declaration) the product of a short polynomial and modular inverse of another short polynomial seems unable to be decomposed; and (2) the product of several short polynomials seems unable to be decomposed. However, the product of several short polynomials is a somewhat short polynomial. Although it cannot be decomposed, it can be used as a modulus to filter the noise. This is the reason that applications of GGH map for public tools of encoding provide insufficient protection to users' secrets. On the other hand, breaking applications of GGH map for public tools of encoding does not mean solving users' secrets. It only means solving "high-order bits of decoding of the product of encodings of users' secrets", a weaker requirement. All of the above form the first principle of our attack. The second principle is that, if one uses GGH map for constructing the instance of WE based on the hardness of 3-exact cover problem, special structure of GGH map can simplify the 3-exact cover problem into a combined 3-exact cover problem.

Although our attack is limited to applications for public tools of encoding, it greatly reduces the value of GGH map. Authors of GGH map [2] presented three variants, which are "asymmetric encoding", "providing zero-test security" and "avoiding principal ideals". We find the first and the second variants never immune to our attack, as long as they are used for MPKE and the instance of WE based on the hardness of 3-exact cover problem. The third variant is under study. A new consideration is removing multiplication commutability, but the application will be greatly limited. For example, the instance of WE based on the hardness of 3-exact cover problem can only use multiplication commutable ideal lattices.

1.3 The Organization

In subsection 1.4 we review recent works related to multilinear map. In section 2 we review GGH map and two applications, MPKE and the instance of WE on 3-exact cover. In section 3 we define special tools for our attack, which are special polynomials used for our modular operations. Also in this section, for the secret of each user, we generate an equivalent secret, which is not small. It is immediate that we obtain an "equivalent secret" of the product of users' secrets, which just is the product of users' equivalent secrets. In section 4 we present modified encoding/decoding. We show how can "high-order bits of decoding of the product of encodings of users' secrets" be solved, so that MPKE is broken. In section 5 we show how to break the instance of WE on 3-exact cover. In this section we first introduce and solve "combined 3-exact cover problem", then solve "high-order bits of decoding of the product of encodings of users' secrets". Section 6 is an auxiliary work. In this section we compute hermite normal form of the specific principal ideal, used for finding special equivalent secret in section 5.

1.4 Related Works

Arita and Handa [5] presented two applications of multilinear maps: group key exchange and witness encryption. Their witness encryption scheme (called AH scheme) has the security claim based on the hardness of Hamilton Cycle problem. The novelty is that they used an asymmetric multilinear map over integer matrices. Bellare and Hoang [6] presented adaptive witness encryption, with stronger security than soundness security, named adaptive soundness security.

Garg et al. [7] presented witness encryption by using indistinguishability obfuscation and Multilinear Jigsaw Puzzle, a simplified variant of multilinear maps. Extractable witness encryption was presented [8–10].

2 GGH map and two applications

2.1 Notations and Definitions

We define the rational numbers by $\mathbb{Q}$ and the integers by $\mathbb{Z}$. We specify that n-dimensional vectors of $\mathbb{Q}^n$ and $\mathbb{Z}^n$ are row vectors. We consider the $2n$'th cyclotomic polynomial ring $R = \mathbb{Z}[X]/(X^n + 1)$, and identify an element $u \in R$ with the coefficient vector of the degree-$(n-1)$ integer polynomial that represents u. In this way, R is identified with the integer lattice $\mathbb{Z}^n$. We also consider the ring $R_q = R/qR = \mathbb{Z}_q[X]/(X^n+1)$ for a (large enough) integer q. Obviously, addition in these rings is done component-wise in their coefficients, and multiplication is polynomial multiplication modulo the ring polynomial $X^n + 1$. In some cases we also consider the ring $\mathbb{K} = \mathbb{Q}[X]/(X^n + 1)$, which is likewise associated with the linear space $\mathbb{Q}^n$. We redefine the operation "mod q" as the follow: if q is an odd, $a(\text{mod } q)$ is within $\{-(q-1)/2, -(q-3)/2, \cdots, (q-1)/2\}$; if q is an even, $a(\text{mod } q)$ is within $\{-q/2, -(q-2)/2, \cdots, (q-2)/2\}$. For $x \in R$, $\langle x \rangle = \{x \cdot u : u \in R\}$ is the principal ideal in R generated by x (alternatively, the sub-lattice of $\mathbb{Z}^n$ corresponding to this ideal). For $x \in R$, $y \in R$, $y(\text{mod } x)$ is such vector: $y(\text{mod } x) = ax$, each entry of a is within [-0.5, 0.5), and $y - y(\text{mod } x) \in \langle x \rangle$.

2.2 Parameter Setting and Map

We secretly sample a short element $g \in R$. Let $\langle g \rangle$ be the principal ideal in R. g itself is kept secret, and no "good" description of $\langle g \rangle$ is made public. Another secret element $z \in R_q$ is chosen at random, and hence is not short.

A public element y is called encoding parameter, and is set as the follow. We secretly sample a short element $a \in R$, and let $y = (1 + ag)z^{-1}(\text{mod } q)$. Public elements $\{x^{(i)}, i = 1, 2\}$ are called randomizers, and are set as the follow. We secretly sample a short element $b^{(i)} \in R$, and let $x^{(i)} = b^{(i)}gz^{-1}(\text{mod } q)$, $i = 1, 2$. Public element p_{zt} is called K-order zero-testing parameter, where $K \geq 3$ is an integer. p_{zt} is set as the follow. We secretly sample a "somewhat small" element $h \in R$, and let $p_{zt} = (hz^k g^{-1})(\text{mod } q)$. Simply speaking, public parameters y and $\{x^{(i)}, i = 1, 2\}$ are encoding tools, while public parameter p_{zt} is decoding tool. $\{g, z, a, \{b^{(i)}, i = 1, 2\}, h\}$ are kept from all users (For another class of applications, encoding tools are hidden).

Suppose a user has a secret $v \in R$, which is a short element. He secretly samples short elements $\{u^{(i)} \in R, i = 1, 2\}$. He computes noised encoding $V = vy + (u^{(1)}x^{(1)} + u^{(2)}x^{(2)})(\text{mod } q)$, where $vy(\text{mod } q)$ and $(u^{(1)}x^{(1)} + u^{(2)}x^{(2)})(\text{mod } q)$ are respectively encoded secret and encoded noise. He publishes V. Then GGH K-linear map includes $K, y, \{x^{(i)}, i = 1, 2\}, p_{zt}$, and all noised encoding V for all users.

We call g grade 1 element, and denote σ as the standard deviation for sampling g. We call $\{a, \{b^{(i)}, i = 1, 2\}\}$ and $\{v, \{u^{(i)}, i = 1, 2\}\}$ grade 2 elements, and denote σ' as the standard deviation for sampling $\{a, \{b^{(i)}, i = 1, 2\}\}$ and $\{v, \{u^{(i)}, i = 1, 2\}\}$. Both σ and σ' are greatly smaller than $\sqrt{q}$, and GGH K-linear map [2] suggests $\sigma' = n\sigma$. Finally, we call h grade 3 element, and take $\sigma'' = \sqrt{q}$ as the standard deviation for sampling h. We say that g, $\{a, \{b^{(i)}, i = 1, 2\}\}$ and $\{v, \{u^{(i)}, i = 1, 2\}\}$ are "very small", and that h is "somewhat small". h can not be taken "very small" for the security concern.

2.3 Application 1: MPKE

Suppose that $K + 1$ users want to generate a common shared key by public discussion. To do so, each user k generates his secret $v^{(k)}$, and publishes the noised encoding $V^{(k)}, k = 1, \cdots, K + 1$. Then each user can use his secret and other users' noised encodings to compute KEY, the common shared key. KEY is high-order bits of any decoded message. For example, user k_0 first computes $v^{(k_0)} p_{zt} \prod_{k \neq k_0} V^{(k)} (\text{mod } q)$, then KEY is high-order bits of $v^{(k_0)} p_{zt} \prod_{k \neq k_0} V^{(k)} (\text{mod } q)$. That is, he first computes

$$v^{(k_0)} p_{zt} \prod_{k \neq k_0} V^{(k)} (\text{mod } q) =$$

$$h(1 + ag)^K g^{-1} \prod_{k=1}^{K+1} v^{(k)} +$$

$$hv^{(k_0)} \sum_{\substack{S \subset \{1, \cdots, K+1\} \\ -\{k_0\}, |S| \geq 1}} (1 + ag)^{K-|S|} g^{|S|-1} \prod_{\substack{k \in \{1, \cdots, K+1\} - \\ \{k_0\} - S}} (v^{(k)}) \prod_{t \in S} (u^{(t,1)} b^{(1)} + u^{(t,2)} b^{(2)}) (\text{mod } q)$$

It is modular sum of two terms, decoded message and decoded noise. Decoded message is

$$h(1 + ag)^K g^{-1} \prod_{k=1}^{K+1} v^{(k)} (\text{mod } q).$$

Decoded noise is

$$hv^{(k_0)} \sum_{\substack{S \subset \{1, \cdots, K+1\} \\ -\{k_0\}, |S| \geq 1}} (1 + ag)^{K-|S|} g^{|S|-1} \prod_{\substack{k \in \{1, \cdots, K+1\} \\ -\{k_0\} - S}} (v^{(k)}) \prod_{t \in S} (u^{(t,1)} b^{(1)} + u^{(t,2)} b^{(2)}).$$

Notice that decoded noise is the sum of $3^K - 1$ terms. For example, $h(1 + ag)^{K-1} b^{(1)} u^{(1,1)} \prod_{k=2}^{K+1} (v^{(k)})$ is a term of the decoded noise. Each term is the product of a "somewhat small" element and several "very small" elements. Therefore decoded noise is "somewhat small", and it can be removed if we only extract high-order bits of $v^{(k_0)} p_{zt} \prod_{k \neq k_0} V^{(k)} (\text{mod } q)$. In other words, KEY actually is high-order bits of decoded message $h(1+ag)^K g^{-1} \prod_{k=1}^{K+1} v^{(k)} (\text{mod } q)$.

2.4 Application 2: the Instance of WE on 3-Exact Cover

3-Exact Cover Problem [3, 18] If we are given a subset of $\{1, 2, \cdots, 3K\}$ including 3 integers, we call it a piece. If we are given a collection of K pieces without intersection, we call it a 3-exact cover of $\{1, 2, \cdots, 3K\}$. The 3-exact cover problem is that, for randomly given $N(K)$ different pieces with a hidden 3-exact cover, find it. It is clear that $1 \leq N(k) \leq C_{3K}^3$. If $N(K) = O(K)$, the 3-exact cover problem is not hard. In this case we can efficiently use subtraction, that is, exclude those pieces which are not included in any 3-exact cover. Generally we take $N(K) = O(K^2)$ to make 3-exact cover problem hard enough.

Encryption The encrypter samples short elements $v^{(1)}, v^{(2)}, \cdots, v^{(3K)} \in R$. He computes the encryption key as the follow. He first computes $v^{(1)}v^{(2)} \cdots v^{(3K)}y^K p_{zt} (\text{mod } q)$, then takes $EKEY$ as its high-order bits. In fact, $EKEY$ is high-order bits of $v^{(1)}v^{(2)} \cdots v^{(3K)}(1 + ag)^K hg^{-1} (\text{mod } q)$. He can uses $EKEY$ and an encryption algorithm to encrypt any plaintext. Then he hides $EKEY$ into pieces as follows. He randomly generates $N(K)$ different pieces of $\{1, 2, \cdots, 3K\}$, with a hidden 3-exact cover called EC. For each piece $\{i_1, i_2, i_3\}$ he computes noised encoding of the product $v^{(i_1)}v^{(i_2)}v^{(i_3)}$, that is, secretly samples short elements $\{u^{(\{i_1,i_2,i_3\},i)} \in R, i = 1, 2\}$, then computes and publishes $V^{\{i_1,i_2,i_3\}} = v^{(i_1)}v^{(i_2)}v^{(i_3)}y + (u^{(\{i_1,i_2,i_3\},1)}x^{(1)} + u^{(\{i_1,i_2,i_3\},2)}x^{(2)}) (\text{mod } q)$.

Decryption The one who knows EC computes the decoding of $\prod_{\{i_1,i_2,i_3\} \in EC} V^{\{i_1,i_2,i_3\}} (\text{mod } q)$, that is, he computes $p_{zt} \prod_{\{i_1,i_2,i_3\} \in EC} V^{\{i_1,i_2,i_3\}} (\text{mod } q)$. Then $EKEY$ is its high-order bits. In other words, $p_{zt} \prod_{\{i_1,i_2,i_3\} \in EC} V^{\{i_1,i_2,i_3\}} (\text{mod } q)$ is modular sum of two terms, the first term is decoded message $v^{(1)}v^{(2)} \cdots v^{(3K)}(1+ ag)^K hg^{-1}$ (mod q), while the second term is decoded noise which doesn't affect high-order bits of $p_{zt} \prod_{\{i_1,i_2,i_3\} \in EC} V^{\{i_1,i_2,i_3\}} (\text{mod } q)$.

3 Weak-DL Attack: Generating Equivalent Secrets

Table 1 is a comparison between processing routines of GGH map and our work. It is a note of our claim that we can achieve same purpose without knowing the secret of any user.

Table 1. Processing routines

GGH map	secrets → encodings → product → decoding → high-order bits
Our work	equivalent secrets → product → modified encoding/decoding → high-order bits

As the start of our attack, we will find equivalent secrets. The method is weak-DL attack [2].

3.1 Generating an Equivalent Secret for One User

We can obtain special decodings $\{Y, X^{(i)}, i = 1, 2\}$, where

$$\begin{aligned}&Y = y^{K-1}x^{(1)}p_{zt}(\text{mod } q) = h(1 + ag)^{K-1}b^{(1)},\\&X^{(i)} = y^{K-2}x^{(i)}x^{(1)}p_{zt}(\text{mod } q) = h(1 + ag)^{K-2}(b^{(i)}g)b^{(1)},\\&i = 1, 2.\end{aligned}$$

Notice that right sides of these equations have no operation "mod q". More precisely, each of $\{Y, X^{(i)}, i = 1, 2\}$ is a factor of a term of decoded noise. For example, $Yu^{(1,1)}\prod_{k=2}^{K+1}(v^{(k)})$ is a term of the decoded noise. Therefore each of $\{Y, X^{(i)}, i = 1, 2\}$ is far smaller than a term of the decoded noise. However, they are not small enough because of the factor h. We say they are "somewhat small", and take them as our tools.

Take the noised encoding V (corresponding to the secret v and unknown $\{u^{(1)}, u^{(2)}\}$), and compute special decoding

$$W = Vy^{K-2}x^{(1)}p_{zt}(\text{mod } q) = vY + (u^{(1)}X^{(1)} + u^{(2)}X^{(2)}).$$

Notice that right side of this equation has no operation "mod q". Then compute

$$W(\text{mod } Y) = \big(u^{(1)}X^{(1)}(\text{mod } Y) + u^{(2)}X^{(2)}(\text{mod } Y)\big)(\text{mod } Y).$$

Step 1 By knowing $W(\text{mod } Y)$ and $\{X^{(1)}(\text{mod } Y), X^{(2)}(\text{mod } Y)\}$, obtain $W' \in \langle X^{(i)}, i = 1, 2\rangle$ such that $W - W'(\text{mod } Y) = 0$. This is quite an easy algebra, and we present the detail in Appendix A. Notice that $W - W'$ is not a short vector. Denote $W' = u'^{(1)}X^{(1)} + u'^{(2)}X^{(2)}$.

Step 2 Compute $v^{(0)} = (W - W')/Y$ (division over the real numbers, with the quotient which is an integer vector). Then

$$\begin{aligned}v^{(0)} &= v + ((u^{(1)}X^{(1)} + u^{(2)}X^{(2)}) - W')/Y\\&= v + ((u^{(1)} - u'^{(1)})X^{(1)} + (u^{(2)} - u'^{(2)})X^{(2)})/Y\\&= v + ((u^{(1)} - u'^{(1)})b^{(1)} + (u^{(2)} - u'^{(2)})b^{(2)})g/(1 + ag).\end{aligned}$$

By considering another fact that g and $1+ag$ are coprime, we have $v^{(0)} - v \in \langle g\rangle$. We call $v^{(0)}$ an equivalent secret of v, and call residual vector $v^{(0)} - v$ the noise. Notice that $v^{(0)}$ is not a short vector.

3.2 Generating an "Equivalent Secret" for the product of Secrets

Suppose that each user k has his secret $v^{(k)}$, and we generate $v^{(0,k)}$, an equivalent secret of $v^{(k)}$, where $k = 1, \cdots, K + 1$. For the product $\prod_{k=1}^{K+1} v^{(k)}$, we have an "equivalent secret" $\prod_{k=1}^{K+1} v^{(0,k)}$, where the noise is $\prod_{k=1}^{K+1} v^{(0,k)} - \prod_{k=1}^{K+1} v^{(k)} \in \langle g\rangle$. Notice that $\prod_{k=1}^{K+1} v^{(0,k)}$ is not a short vector.

4 Modified Encoding/Decoding

In this section we transform $\prod_{k=1}^{K+1} v^{(0,k)}$ by our modified encoding/decoding. The procedure has three steps, which are multiplication by Y, mod $X^{(1)}$ operation, and mod q multiplication by $y(x^{(1)})^{-1}$(or by $Y(X^{(1)})^{-1}$). Denote $\eta = \prod_{k=1}^{K+1} v^{(0,k)}$. Then $\eta = \prod_{k=1}^{K+1} v^{(k)} + \xi g$, where $\xi \in R$.

Step 1 Compute $\eta' = Y\eta$. By noticing that Y is a multiple of $b^{(1)}$, we have a fact that $\eta' = Y\prod_{k=1}^{K+1} v^{(k)} + \xi' b^{(1)} g$, where $\xi' \in R$.

Step 2 Compute $\eta'' = \eta'(\text{mod } X^{(1)})$. There are 3 facts as follows.

(1) $\eta'' = Y\prod_{k=1}^{K+1} v^{(k)} + \xi'' b^{(1)} g$, where $\xi'' \in R$. Notice that η'' is the sum of η' and a multiple of $X^{(1)}$, and that $X^{(1)}$ is a multiple of $b^{(1)}g$.

(2) η'' has the size similar to that of $\sqrt{n}X^{(1)}$. In other words, η'' is smaller than one term of decoded noise. Notice standard deviations for sampling various variables.

(3) $Y\prod_{k=1}^{K+1} v^{(k)}$ has the size similar to that of one term of decoded noise.

Above 3 facts result in a new fact that $\xi'' b^{(1)} g = \eta'' - Y\prod_{k=1}^{K+1} v^{(k)}$ has the size similar to that of one term of decoded noise.

Step 3 Compute $\eta''' = y(x^{(1)})^{-1}\eta''(\text{mod } q)$. There are 3 facts as follows.

(1) $\eta''' = (h(1+ag)^K g^{-1})\prod_{k=1}^{K+1} v^{(k)} + \xi''(1+ag)(\text{mod } q)$. Notice fact (1) of Step 2, and notice the definitions of Y and $X^{(1)}$.

(2) $\xi''(1+ag)$ has the size similar to that of one term of decoded noise. In other words, $\xi''(1+ag)$ is smaller than decoded noise. This fact is clear by noticing that $\xi'' b^{(1)} g$ has the size similar to that of one term of decoded noise, and by noticing that $1+ag$ and $b^{(1)}g$ have similar size.

(3) $(h(1+ag)^K g^{-1})\prod_{k=1}^{K+1} v^{(k)}(\text{mod } q)$ is decoded message, therefore its high-order bits are what we want to obtain.

Above 3 facts result in a new fact that η''' is modular sum of decoded message and a new decoded noise which is smaller than original decoded noise. Therefore high-order bits of η''' are what we want to obtain. MPKE has been broken. More important is that K-GMDDH assumption (Assumption 5.1 of [11]) is negated.

5 Breaking the Instance of WE Based on the Hardness of 3-Exact Cover Problem

Our modified encoding/decoding can not directly break the instance of WE based on the hardness of 3-exact cover problem, because the 3-exact cover is hidden. In this section we show that special structure of GGH map can simplify the 3-exact cover problem into a combined 3-exact cover problem, then show how to use a combined exact cover to break the instance.

5.1 Combined 3-Exact Cover Problem: Definition and Solution

Suppose we are given $N(K) = O(K^2)$ different pieces of $\{1, 2, \cdots, 3K\}$. A subset $\{i_1, i_2, i_3\}$ of $\{1, 2, \cdots, 3K\}$ is called a combined piece, if

(1) $\{i_1, i_2, i_3\}$ is not a piece;

(2) $\{i_1, i_2, i_3\} = \{j_1, j_2, j_3\} \cup \{k_1, k_2, k_3\} - \{l_1, l_2, l_3\}$;

(3) $\{j_1, j_2, j_3\}$, $\{k_1, k_2, k_3\}$ and $\{l_1, l_2, l_3\}$ are pieces.

(Then $\{j_1, j_2, j_3\}$ and $\{k_1, k_2, k_3\}$ don't intersect, and $\{j_1, j_2, j_3\} \cup \{k_1, k_2, k_3\} \supset \{l_1, l_2, l_3\}$).

A subset $\{i_1, i_2, i_3\}$ of $\{1, 2, \cdots, 3K\}$ is called a second-order combined piece, if

(1) $\{i_1, i_2, i_3\}$ is neither a piece nor a combined piece;

(2) $\{i_1, i_2, i_3\} = \{j_1, j_2, j_3\} \cup \{k_1, k_2, k_3\} - \{l_1, l_2, l_3\}$;

(3) $\{j_1, j_2, j_3\}$, $\{k_1, k_2, k_3\}$ and $\{l_1, l_2, l_3\}$ are pieces or combined pieces.

K pieces or combined pieces or second-order combined pieces without intersection are called a combined 3-exact cover of $\{1, 2, \cdots, 3K\}$. The combined 3-exact cover problem is that, for randomly given $N(K) = O(K^2)$ different pieces, find a combined 3-exact cover. More specifically, we take $N(K) = K^2$ without loss of generality. We will show that the combined 3-exact cover problem is never hard.

Obtaining Combined Pieces Suppose that K^2 pieces are sufficiently random distributed, and in them there is a hidden 3-exact cover. We take $P(E)$ as the probability of the event E, and $P\big(E\big|E'\big)$ as the conditional probability of E under the condition E'. Arbitrarily take a subset $\{i_1, i_2, i_3\}$ which is not a piece. In Appendix B we show that $P(\{i_1, i_2, i_3\}$ is not a combined piece) $\approx e^{-2}$. Now we construct all combined pieces from K^2 pieces, then we have a result: there are about $(1 - e^{-2})C_{3K}^3$ different subsets of $\{1, 2, \cdots, 3K\}$, each including 3 elements, which are pieces or combined pieces.

Obtaining Second-Order Combined Pieces There are about $e^{-2}C_{3K}^3$ different subsets of $\{1, 2, \cdots, 3K\}$, each including 3 elements, which are neither pieces nor combined pieces. Arbitrarily take one subset $\{i_1, i_2, i_3\}$ from them. By similar deduction procedure to Appendix B, we can show that $P(\{i_1, i_2, i_3\}$ is not a second-order combined piece) is negatively exponential in K. Now we construct all second-order combined pieces from $(1 - e^{-2})C_{3K}^3$ pieces or combined pieces, then we are almost sure to have a result: all C_{3K}^3 different subsets of $\{1, 2, \cdots, 3K\}$, each including 3 elements, are pieces or combined pieces or second-order combined pieces. Therefore the combined 3-exact cover problem is solved.

5.2 Positive/Negative Factors

Definition 1. *Take a fixed combined 3-exact cover. Take an element $\{i_1, i_2, i_3\}$ of this combined 3-exact cover.*

(1) If $\{i_1, i_2, i_3\}$ is a piece, we count it as a positive factor.

(2) If $\{i_1, i_2, i_3\}$ is a combined piece, $\{i_1, i_2, i_3\} = \{j_1, j_2, j_3\} \cup \{k_1, k_2, k_3\} - \{l_1, l_2, l_3\}$, we count pieces $\{j_1, j_2, j_3\}$ and $\{k_1, k_2, k_3\}$ as positive factors, and count the piece $\{l_1, l_2, l_3\}$ as a negative factor.

(3) Suppose $\{i_1, i_2, i_3\}$ is a second-order combined piece, $\{i_1, i_2, i_3\} = \{j_1, j_2, j_3\} \cup \{k_1, k_2, k_3\} - \{l_1, l_2, l_3\}$, where $\{j_1, j_2, j_3\}$, $\{k_1, k_2, k_3\}$ and $\{l_1, l_2, l_3\}$ are pieces or combined pieces.

(3.1) If $\{j_1, j_2, j_3\}$ is a piece, we count it as a positive factor; if $\{j_1, j_2, j_3\}$ is a combined piece, we count 2 positive factors corresponding to it as positive factors, and the negative factor corresponding to it as a negative factor.

(3.2) Similarly, if $\{k_1, k_2, k_3\}$ is a piece, we count it as a positive factor; if $\{k_1, k_2, k_3\}$ is a combined piece, we count 2 positive factors corresponding to it as positive factors, and the negative factor corresponding to it as a negative factor.

(3.3) Oppositely, if $\{l_1, l_2, l_3\}$ is a piece, we count it as a negative factor; if $\{l_1, l_2, l_3\}$ is a combined piece, we count 2 positive factors corresponding to it as negative factors, and the negative factor corresponding to it as a positive factor.

Positive and negative factors are pieces. All positive factors form a collection, and all negative factors form another collection (notice that we use the terminology "collection" rather than "set", because it is possible that one piece is counted several times). Take CPF as the collection of positive factors, NPF as the number of positive factors. Take CNF as the collection of negative factors, NNF as the number of negative factors. Notice that some pieces may be counted repeatedly. It is easy to see that $NPF - NNF = K$. On the other hand, from C_{3K}^3 different subsets of $\{1, 2, \cdots, 3K\}$, there are K^2 different pieces, about $(1 - e^{-2})C_{3K}^3 - K^2$ different combined pieces, and about $e^{-2}C_{3K}^3$ different second-order combined pieces. Each piece is a positive factor, each combined piece is attached by 2 positive factors and a negative factor, each second-order combined piece is attached by at most 5 positive factors and 4 negative factors. Therefore, for a randomly chosen combined 3-exact cover, it is almost sure that $NPF \leq 3K$, resulting in $NNF \leq 2K$.

5.3 Our Construction

Randomly take a combined 3-exact cover. Obtain CPF, the collection of positive factors, and CNF, the collection of negative factors. For a positive factor $pf = \{i_1, i_2, i_3\}$, we denote $v^{(pf)} = v^{(i_1)}v^{(i_2)}v^{(i_3)}$ as the secret of pf, and $v'^{(pf)}$ as the equivalent secret of $v^{(pf)}$ obtained in subsection 3.1. Similarly we denote $v^{(nf)}$ and $v'^{(nf)}$ for a negative factor nf. Denote $PPF = \prod_{pf \in CPF} v'^{(pf)}$ as the product of equivalent secrets of all positive factors. Denote $PNF = \prod_{nf \in CNF} v'^{(nf)}$ as the product of equivalent secrets of all negative factors. Denote $PTS = \prod_{k=1}^{3K} v^{(k)}$ as the product of true secrets. The first clear equation is

$\prod_{pf\in CPF} v^{(pf)} = PTS \times \prod_{nf\in CNF} v^{(nf)}$. Then we have

Proposition 1

(1) $PPF - \prod_{pf\in CPF} v^{(pf)} \in \langle g\rangle$.
(2) $PNF - \prod_{nf\in CNF} v^{(nf)} \in \langle g\rangle$.
(3) $PPF - PNF \times PTS \in \langle g\rangle$.

Proof. By considering subsection 3.1, we know that

(1) $PPF = \prod_{pf\in CPF} v^{(pf)} + \beta_{PF}$, where $\beta_{PF} \in \langle g\rangle$.
(2) $PNF = \prod_{nf\in CNF} v^{(nf)} + \beta_{NF}$, where $\beta_{NF} \in \langle g\rangle$.

On the other hand, (3) is true from

$$\prod_{pf\in CPF} v^{(pf)} = PTS \times \prod_{nf\in CNF} v^{(nf)}.$$

Proposition 1 is proved. □

Maybe it is hoped to solve PTS. However, we can not filter off β_{PF} and β_{NF}, because no "good" description of $\langle g\rangle$ is made public. Fortunately we don't need to solve PTS for breaking the instance. We only need to find an equivalent secret of PTS, without caring the size of the equivalent secret. Then we can filter decoded noise much smaller by our modified encoding/decoding. Proposition 2 describes the shape of the equivalent secret of PTS, under an assumption.

Proposition 2

(1) If PTS' is an equivalent secret of PTS, then $PPF - PNF \times PTS' \in \langle g\rangle$.
(2) Assume that PNF and g are coprime. If $PPF - PNF \times PTS' \in \langle g\rangle$, then PTS' is an equivalent secret of PTS.

Proof. (1) is clear by considering (3) of Proposition 1. If $PPF - PNF \times PTS' \in \langle g\rangle$, then $PNF \times (PTS' - PTS) \in \langle g\rangle$. According to our assumption, we have $(PTS' - PTS) \in \langle g\rangle$, (2) is proved. □

Now we want to find an equivalent secret of PTS. Under our assumption, we only need to find a vector $PTS' \in R$ such that $PPF - PNF \times PTS' \in \langle g\rangle$, without caring the size of PTS'. To do so we only need to obtain "bad" description of $\langle g\rangle$. That is, we only need to obtain Hermite normal form $G = \begin{bmatrix} G_0 & & & \\ G_1 & 1 & & \\ \vdots & & \ddots & \\ G_{n-1} & & & 1 \end{bmatrix}$, where each row of G is an element of $\langle g\rangle$, G_0 is absolute value of the determinant of the matrix $\begin{bmatrix} g_0 & g_1 & \cdots & g_{n-1} \\ -g_{n-1} & g_0 & \cdots & g_{n-2} \\ \vdots & \vdots & \ddots & \vdots \\ -g_1 & -g_2 & \cdots & g_0 \end{bmatrix}$, and $G_i (\text{mod } G_0) = G_i$ for $i = 1, \cdots, n-1$. Computing the Hermite normal form G is not a difficult task, and in the next section we will present our method for doing so. After obtaining

G, the condition $PPF - PNF \times PTS' \in \langle g \rangle$ is transformed into an equivalent condition

$$PPF \times G^{-1} - PTS' \times \overline{PNF} \times G^{-1} \in R,$$

where G^{-1} is the inverse matrix of G, and

$$\overline{PNF} = \begin{bmatrix} PNF_0 & PNF_1 & \cdots & PNF_{n-1} \\ -PNF_{n-1} & PNF_0 & \cdots & PNF_{n-2} \\ \vdots & \vdots & \ddots & \vdots \\ -PNF_1 & -PNF_2 & \cdots & PNF_0 \end{bmatrix}.$$

Take each entry of $PPF \times G^{-1}$ and $\overline{PNF} \times G^{-1}$ as the form of reduced fraction, and take lcm as the least common multiple of all denominators, then the condition is transformed into another equivalent condition

$$\begin{aligned} &(lcm \times PPF \times G^{-1})(\text{mod } lcm) \\ &= PTS' \times (lcm \times \overline{PNF} \times G^{-1})(\text{mod } lcm). \end{aligned}$$

This is a linear equation modular lcm, and it is easy to obtain a solution PTS'. After that we take our modified encoding/decoding, just same as in section 4. Denote $\eta = PTS'$. Compte $\eta' = Y\eta$. Compute $\eta'' = \eta'(\text{mod } X^{(1)})$. Compute $\eta''' = y(x^{(1)})^{-1}\eta''(\text{mod } q)$. Then high-order bits of η''' are what we want to obtain. The instance has been broken.

A question left is whether the assumption "PNF and g are coprime" is a nonnegligible case. It means that g and each factor of PNF are coprime. The answer is seemingly yes. A test which we haven't made is that we take two different combined 3-exact covers, so that we obtain two different values of PNF. If they finally get same high-order bits of η''', we can believe the assumption true for two values of PNF.

6 Computing Hermite Normal Form of $\langle g \rangle$

For a principal ideal $\langle g' \rangle$, we call the determinant of $\begin{bmatrix} g'_0 & g'_1 & \cdots & g'_{n-1} \\ -g'_{n-1} & g'_0 & \cdots & g'_{n-2} \\ \vdots & \vdots & \ddots & \vdots \\ -g'_1 & -g'_2 & \cdots & g'_0 \end{bmatrix}$ corresponding determinant of $\langle g' \rangle$. We use the definition of parallel piped [19]. For a vector $\alpha \in R$, we call the set $PP(\alpha) = \{z \in R : z(\text{mod } \alpha) = z\}$ parallel piped of α.

6.1 Two Facts

We have $\{Y, X^{(i)}, i = 1, 2\}$, therefore we can obtain hermite normal forms of the principal ideals $\{\langle Y \rangle, \langle X^{(i)} \rangle, i = 1, 2\}$.

Suppose hermite normal form of the principal ideal $\langle g' \rangle$ is $\begin{bmatrix} G'_0 & & & \\ G'_1 & 1 & & \\ \vdots & & \ddots & \\ G'_{n-1} & & & 1 \end{bmatrix}$, $g \in R$ is a factor of g', and absolute value of corresponding determinant of $\langle g \rangle$ is G_0. Then hermite normal form of the principal ideal $\langle g \rangle$ is $\begin{bmatrix} G_0 & & & \\ G'_1(\text{mod } G_0) & 1 & & \\ \vdots & & \ddots & \\ G'_{n-1}(\text{mod } G_0) & & & 1 \end{bmatrix}$.

6.2 Computing Hermite Normal Form of $\langle h(1+ag)^{K-2}b^{(1)} \rangle$

We take a trivial assumption that $1+ag$ and $b^{(1)}g$ are coprime.

Step 1 By using $\{Y, (-Y_{n-1}, Y_0, \cdots, Y_{n-2}), \cdots, (-Y_1, \cdots, -Y_{n-1}, Y_0)\}$ as the basis, Gaussian sample Z, with sufficiently large deviation.

Step 2 Compute $Z' = Z(\text{mod } X^{(1)})$. Then Z' is uniformly distributed over the intersection area $\langle h(1+ag)^{K-2}b^{(1)} \rangle \cap PP(X^{(1)})$. Algebra and Gaussian sampling theory have proved this result.

Step 3 Compute absolute value of corresponding determinant of $\langle Z' \rangle$.

Step 4 Repeat Step 1∼3 polynomially many times, so that we obtain polynomially many absolute values of corresponding determinant.

Step 5 Compute the greatest common divisor of these polynomially many absolute values. Then the greatest common divisor should be absolute value of corresponding determinant of $\langle h(1+ag)^{K-2}b^{(1)} \rangle$. By considering a fact stated in last subsection, we obtain hermite normal form of $\langle h(1+ag)^{K-2}b^{(1)} \rangle$.

6.3 Computing Hermite Normal Form of $\langle h(1+ag)^{K-2}b^{(1)}g \rangle$

We take a trivial assumption that $b^{(1)}$ and $b^{(2)}$ are coprime. The procedure is similar to last subsection.

Step 1 By using $\{X^{(2)}, (-X^{(2)}_{n-1}, X^{(2)}_0, \cdots, X^{(2)}_{n-2}), \cdots, (-X^{(2)}_1, \cdots, -X^{(2)}_{n-1}, X^{(2)}_0)\}$ as the basis, Gaussian sample Z, with sufficiently large deviation.

Step 2 Compute $Z' = Z(\text{mod } X^{(1)})$. Then Z' is uniformly distributed over the intersection area $\langle h(1+ag)^{K-2}b^{(1)}g \rangle \cap PP(X^{(1)})$.

Step 3 Compute absolute value of corresponding determinant of $\langle Z' \rangle$.

Step 4 Repeat Step 1∼3 polynomially many times, so that we obtain polynomially many absolute values of corresponding determinant.

Step 5 Compute the greatest common divisor of these polynomially many absolute values. Then the greatest common divisor should be absolute value of corresponding determinant of $\langle h(1+ag)^{K-2}b^{(1)}g \rangle$, therefore, we obtain hermite normal form of $\langle h(1+ag)^{K-2}b^{(1)}g \rangle$.

6.4 Obtaining Hermite Normal Form of $\langle g \rangle$

Divide absolute value of corresponding determinant of $\langle h(1+ag)^{K-2}b^{(1)}g\rangle$ by absolute value of corresponding determinant of $\langle h(1+ag)^{K-2}b^{(1)}\rangle$. Then we obtain absolute value of corresponding determinant of $\langle g\rangle$, therefore we obtain hermite normal form of $\langle g\rangle$.

Acknowledgments. We are very grateful for helps and suggestions from authors of GGH map.

References

1. Boneh, D., Silverberg, A.: Applications of Multilinear Forms to Cryptography. Contemporary Mathematics. 324, 71–90 (2003)
2. Garg, S., Gentry, C., Halevi, S.: Candidate Multilinear Maps from Ideal Lattices. In: Johansson, T., Nguyen, P.Q. (ed.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 181–184. Springer, Heidelberg (2013)
3. Garg, S., Gentry, C., Sahai, A., Waters, B.: Witness Encryption and its Applications. In: STOC (2013)
4. Gentry, C., Lewko, A., Waters, B.: Witness Encryption from Instance Independent Assumptions. In: Garay, J.A., Gennaro, R. (ed.) CRYPTO 2014. LNCS, vol. 8616, pp. 426–443. Springer, Heidelberg (2014)
5. Arita, S., Handa, S.: Two Applications of Multilinear Maps: Group Key Exchange and Witness Encryption. In: Proceedings of the 2nd ACM workshop on ASIA public-key cryptography(ASIAPKC '14). ACM, New York, NY, USA, pp. 13–22 (2014)
6. Bellare, M., Hoang V.T.: Adaptive Witness Encryption and Asymmetric Password-Based Cryptography. Cryptology ePrint Archive, Report 2013/704 (2013)
7. Garg, S., Gentry, C., Halevi, S., Raykova, M., Sahai, A., Waters, B.: Candidate Indistinguishability Obfuscation and Functional Encryption for all Circuits. In: FOCS (2013)
8. Goldwasser, S., Kalai, Y.T., Popa, R.A., Vaikuntanathan, V., Zeldovich, N.: How to Run Turing Machines on Encrypted Data. In: Canetti, R., Garay, J.A. (ed.) CRYPTO 2013, Part II. LNCS, vol. 8043, pp. 536–553. Springer, Heidelberg (2013)
9. Garg, S., Gentry, C., Halevi, S., Wichs, D.: On the Implausibility of Differing-Inputs Obfuscation and Extractable Witness Encryption with Auxiliary Input. In: Garay, J.A., Gennaro, R. (ed.) CRYPTO 2014, Part I. LNCS, vol. 8616, pp. 518–535. Springer, Heidelberg (2014)
10. Boyle, E., Chung, K.-M., Pass, R.: On Extractability (a.k.a. Differing-Input) Obfuscation. In: Lindell, Y. (ed.) TCC 2014. LNCS, vol. 8349, pp. 52–73. Springer, Heidelberg (2014)
11. Garg, S., Gentry, C., Halevi, S., Sahai, A., Waters, B.: Attribute-Based Encryption for Circuits from Multilinear Maps. In: Canetti, R., Garay, J.A. (ed.) CRYPTO 2013, Part II. LNCS, vol. 8043, pp. 479–499. Springer, Heidelberg (2013)
12. Langlois, A., Stehl, D., Steinfeld, R.: GGHLiteMore Efficient Multilinear Maps from Ideal Lattices. In: Nguyen, P.Q., Oswald, E. (ed.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 239–256. Springer, Heidelberg (2014)
13. Coron, J.-S., Lenpoint, T., Tibouchi, M.: Practical Multilinear Maps over the Integers. In: Canetti, R., Garay, J.A. (ed.) CRYPTO 2013, Part I. LNCS, vol. 8042, pp. 476–493. Springer, Heidelberg (2013)
14. Cheon, J.H., Han, K., Lee, C., Ryu, H., Stehl. D: Cryptanalysis of the Multilinear Map over the Integers. Cryptology ePrint Archive, Report 2014/906 (2014)
15. Gentry, C., Halevi, S., Maji, H.K., Sahai, A.: Zeroizing without Zeroes: Cryptanalyzing Multilinear Maps without Encodings of Zero. Cryptology ePrint Archive, Report 2014/929 (2014)
16. Boneh, D., Wu, D.J., Zimmerman, J.: Immunizing Multilinear Maps Against Zeroizing Attacks. Cryptology ePrint Archive, Report 2014/930 (2014)
17. Coron, J.-S., Lepoint, T., Tibouchi, M.: Cryptanalysis of Two Candidate Fixes of Multilinear Maps over the Integers. Cryptology ePrint Archive, Report 2014/975 (2014)

18. Goldreich, O.: Computational Complexity: a Conceptual Perspective. Cambridge University Press, New York, NY, USA, 1 edition (2008)

19. Nguyen, P.Q., Regev, O.: Learning a Parallel Piped: Cryptanalysis of GGH and NTRU Signatures. Journal of Cryptology. 22(2), 139–160 (2009)

Appendix

A

Suppose $W(\text{mod } Y) = W'''Y$, $X^{(1)}(\text{mod } Y) = X'^{(1)}Y, X^{(2)}(\text{mod } Y) = X'^{(2)}Y$. We want to obtain a solution $u'^{(i)} \in R$, $i = 1, 2$, such that $W'''Y = (u'^{(1)}X'^{(1)} + u'^{(2)}X'^{(2)})Y(\text{mod } Y)$. First, the equation has solution, because $\{u^{(i)} \in R, i = 1, 2\}$ is a solution. Second, the equation can be modified as an equivalent e-quation $W''' = (u'^{(1)}X'^{(1)} + u'^{(2)}X'^{(2)})(\text{mod } 1)$. Third, take each entry of W''', $X'^{(1)}, X'^{(2)}$ as the form of reduced fraction, and take LCM as the least common multiple of all denominators, then the equation can be modified as an equivalent equation, which is a linear equation modular LCM:

$$(LCM)W''' = (u'^{(1)}((LCM)X'^{(1)}) + u'^{(2)}((LCM)X'^{(2)}))(\text{mod } (LCM)).$$

B

Arbitrarily take a subset $\{i_1, i_2, i_3\}$ which is not a piece. We will compute $P(\{i_1, i_2, i_3\}$ is not a combined piece$)$. We define $N\{i_1, i_2\}$ as the number of those pieces which include $\{i_1, i_2\}$. Similarly we define $N\{i_1, i_3\}$ and $N\{i_2, i_3\}$. We have

Proposition 3

$$\begin{aligned}
&P(\{i_1, i_2, i_3\} \text{ is not a combined piece}) = \\
&P(N\{i_1, i_2\} = 0, N\{i_1, i_3\} = 0, N\{i_2, i_3\} = 0) + \\
&P(N\{i_1, i_2\} > 0, \text{ and } \{i_1, i_2, i_3\} \text{ is not a combined piece}) + \\
&P(N\{i_1, i_2\} = 0, N\{i_1, i_3\} > 0, \text{ and } \{i_1, i_2, i_3\} \text{ is not a combined piece}) + \\
&P(N\{i_1, i_2\} = 0, N\{i_1, i_3\} = 0, N\{i_2, i_3\} > 0, \text{ and } \{i_1, i_2, i_3\} \text{ is not a combined piece}). \square
\end{aligned}$$

$N\{i_1, i_2\}$ has the binomial distribution $b\Big(K^2, \dfrac{3K-2}{C_{3K}^3}\Big)$. So that

Proposition 4 $P(N\{i_1, i_2\} = 0) = \Big(1 - \dfrac{3K-2}{C_{3K}^3}\Big)^{K^2} \approx e^{-2/3}$. □

Now suppose $N\{i_1, i_2\} = u > 0$ and denote $\{\{i_1, i_2, t_1\}, \{i_1, i_2, t_2\}, \cdots, \{i_1, i_2, t_u\}\}$ as u different pieces. From them we randomly take a piece $\{i_1, i_2, t\}$. Take $N\{i_3; i_1, i_2, t\}$ as the number of those pieces which include i_3 and do not include i_1, i_2 and t (In other words, $N\{i_3; i_1, i_2, t\}$ is the number of those pieces which include i_3 and do not intersect with $\{i_1, i_2, t\}$). $N\{i_3; i_1, i_2, t\}$ has the binomial distribution $b\Big(K^2 - u, \dfrac{C_{3K-4}^2}{C_{3K}^3}\Big)$, and $P(N\{i_3; i_1, i_2, t\} = 0) = \Big(1 - \dfrac{C_{3K-4}^2}{C_{3K}^3}\Big)^{K^2-u}$.

Under the condition $N\{i_1, i_2\} = u$, the conditional probability that $\{i_1, i_2, i_3\}$ is not a combined piece is

$$\begin{aligned}
&P\big(\{i_1, i_2, i_3\} \text{ is not a combined piece}\big|N\{i_1, i_2\} = u\big)\\
&= P\big(N\{i_3; i_1, i_2, t_1\} = 0, N\{i_3; i_1, i_2, t_2\} = 0, \cdots, N\{i_3; i_1, i_2, t_u\} = 0\big|N\{i_1, i_2\} = u\big)\\
&< P\big(N\{i_3; i_1, i_2, t_1\} = 0\big|N\{i_1, i_2\} = u\big)\\
&= P\big(N\{i_3; i_1, i_2, t_1\} = 0\big)\\
&= \Big(1 - \frac{C_{3K-4}^2}{C_{3K}^3}\Big)^{K^2-u}.
\end{aligned}$$

Then we obtain

$$\begin{aligned}
&P\big(N\{i_1, i_2\} > 0, \text{ and } \{i_1, i_2, i_3\} \text{ is not a combined piece}\big)\\
&= \sum_{u=1}^{K^2} P\big(N\{i_1, i_2\} = u\big) \cdot P\big(\{i_1, i_2, i_3\} \text{ is not a combined piece}\big|N\{i_1, i_2\} = u\big)\\
&< \sum_{u=1}^{K^2} C_{K^2}^u \Big(\frac{3K-2}{C_{3K}^3}\Big)^u \Big(1 - \frac{3K-2}{C_{3K}^3}\Big)^{K^2-u} \Big(1 - \frac{C_{3K-4}^2}{C_{3K}^3}\Big)^{K^2-u}\\
&= \Big(1 - \Big(1 - \frac{3K-2}{C_{3K}^3}\Big) \cdot \frac{C_{3K-4}^2}{C_{3K}^3}\Big)^{K^2} - \Big(1 - \frac{3K-2}{C_{3K}^3}\Big)^{K^2} \Big(1 - \frac{C_{3K-4}^2}{C_{3K}^3}\Big)^{K^2}\\
&\approx \Big(1 - \frac{1}{K}\Big)^{K^2} - \Big(1 - \frac{1}{K}\Big)^{K^2}\\
&= 0.
\end{aligned}$$

Therefore we have

Proposition 5

$$\begin{aligned}
&P\big(N\{i_1, i_2\} > 0, \text{ and } \{i_1, i_2, i_3\} \text{ is not a combined piece}\big) \approx 0.\\
&P\big(N\{i_1, i_2\} = 0, N\{i_1, i_3\} > 0, \text{ and } \{i_1, i_2, i_3\} \text{ is not a combined piece}\big) \approx 0.\\
&P\big(N\{i_1, i_2\} = 0, N\{i_1, i_3\} = 0, N\{i_2, i_3\} > 0, \text{ and } \{i_1, i_2, i_3\} \text{ is not a combined piece}\big) \approx 0.\square
\end{aligned}$$

By combining Proposition 3∼5, we have

Proposition 6 If $\{N\{i_1, i_2\} = 0\}$, $\{N\{i_1, i_3\} = 0\}$, $\{N\{i_2, i_3\} = 0\}$ are independent each other, then

$$\begin{aligned}
&P\big(\{i_1, i_2, i_3 \text{ is not a combined piece}\big)\\
&\approx P\big(N\{i_1, i_2\} = 0, N\{i_1, i_3\} = 0, N\{i_2, i_3\} = 0\big)\\
&\approx \big(e^{-2/3}\big)^3\\
&= e^{-2}.
\end{aligned}$$

□

In practical parameter setting, $\{N\{i_1, i_2\} = 0\}$, $\{N\{i_1, i_3\} = 0\}$, $\{N\{i_2, i_3\} = 0\}$ are not independent each other. They are usually negatively correlated, that is, larger value of $N\{i_1, i_2\}$ tends to companion smaller value of $N\{i_1, i_3\}$, larger values of $N\{i_1, i_2\}$ and $N\{i_1, i_3\}$ tend to companion smaller value of $N\{i_2, i_3\}$, and vice-versa. This negative correlation feature makes $P\big(N\{i_1, i_2\} = 0, N\{i_1, i_3\} = 0, N\{i_2, i_3\} = 0\big)$ even smaller than e^{-2}. This fact is not important for our attack, and we can roughly take e^{-2} as the probability of the event $\{\{i_1, i_2, i_3\}$ is not a combined piece$\}$.