Cryptanalysis of GGH15 Multilinear Maps

Jean-Sébastien Coron

University of Luxembourg

October 26, 2015

Abstract. We describe a cryptanalysis of the GGH15 multilinear maps. Our attack breaks the multipartite key-agreement protocol by generating an equivalent user private key.

1 Introduction

We describe a cryptanalysis of the GGH15 graph-induced multilinear maps from lattices [GGH15]. Our attack breaks the multipartite key-agreement protocol by generating an equivalent user private key. Our attack proceeds in two steps: in the first step, we express the secret exponent of one user as a linear combination of some other secret exponents corresponding to public encodings, using a variant of the Cheon *et al.* attack [CHL$^+$15]. This does not immediately break the multipartite key-agreement protocol because the coefficients of the linear combination are not small. In the second step, we use the previous linear combination to derive an encoding equivalent to the user private encoding, by correcting the error resulting from the large coefficients of the linear combination.

2 The GGH15 Scheme

We briefly recall the GGH15; we refer to [GGH15] for a full description. In the following we only consider the commutative variant from [GGH15, Section 3.2]; namely this commutative variant must be used in the multipartite key-agreement protocol from [GGH15, Section 5.1].

2.1 GGH15 Multilinear Maps

We work over polynomial rings $R = \mathbb{Z}[x]/(f(X))$ and $R_q = R/qR$ for some degree n irreducible integer polynomial $f(X) \in \mathbb{Z}[x]$ and an integer q. An encoding of a plaintext element $s \in R$ relative to path $u \to v$ is a small matrix $\boldsymbol{D} \in R^{m\times m}$ such that:

$$\boldsymbol{A}_u \cdot \boldsymbol{D} = s \cdot \boldsymbol{A}_v + \boldsymbol{E} \pmod q$$

where $\boldsymbol{A}_u$ and $\boldsymbol{V}_v$ are row vectors of dimension m over R_q, and $\boldsymbol{E}$ is a small row error vector of dimension m. Only small plaintext elements $s \in R$ are encoded.

Two encodings $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$ relative to path $u \to v$ and $v \to w$ can be multiplied to get an encoding relative to path $u \to w$. Namely given:

$$\begin{aligned}\boldsymbol{A}_u \cdot \boldsymbol{C}_1 &= s_1 \cdot \boldsymbol{A}_v + \boldsymbol{E}_1 \pmod q\\ \boldsymbol{A}_v \cdot \boldsymbol{C}_2 &= s_2 \cdot \boldsymbol{A}_w + \boldsymbol{E}_2 \pmod q\end{aligned}$$

we obtain:

$$\begin{aligned}\boldsymbol{A}_u \cdot \boldsymbol{C}_1 \cdot \boldsymbol{C}_2 &= (s_1 \cdot \boldsymbol{A}_v + \boldsymbol{E}_1) \cdot \boldsymbol{C}_2 \pmod q \\ &= s_1 \cdot s_2 \cdot \boldsymbol{A}_w + s_1 \cdot \boldsymbol{E}_2 + \boldsymbol{E}_1 \cdot \boldsymbol{C}_2 \pmod q \\ &= s_1 \cdot s_2 \cdot \boldsymbol{A}_w + \boldsymbol{E}'\end{aligned}$$

Since s_1, $\boldsymbol{E}_1$, $\boldsymbol{E}_2$ and $\boldsymbol{C}_2$ have small coefficients, the error vector $\boldsymbol{E}'$ still has small coefficients (compared to q), and therefore the product $\boldsymbol{C}_1 \cdot \boldsymbol{C}_2$ is an encoding of $s_1 \cdot s_2$ for the path $u \to w$.

Finally, given an encoding $\boldsymbol{C}$ relative to path $u \to w$ and the vector $\boldsymbol{A}_u$, extraction works by computing the high-order bits of $\boldsymbol{A}_u \cdot \boldsymbol{C}$. Namely we have:

$$\boldsymbol{A}_u \cdot \boldsymbol{C} = s \cdot \boldsymbol{A}_w + \boldsymbol{E} \pmod q$$

for some small $\boldsymbol{E}$, and therefore the high-order bits of $\boldsymbol{A}_u \cdot \boldsymbol{C}$ only depend on the secret exponent s.

2.2 The GGH15 Multipartite Key-Agreement

We briefly recall the multipartite key-agreement protocol from [GGH15, Section 5.1]. We consider the protocol with k users. Each user i has a directed path of vectors $\boldsymbol{A}_{i,1}, \ldots, \boldsymbol{A}_{i,k+1}$, all sharing the same end-point $\boldsymbol{A}_0 = \boldsymbol{A}_{i,k+1}$. The i-th user will use the resulting chain to extract the session key. Each user i has a secret exponent s_i. Each secret exponent s_i will be encoded in each of the j chains; the encodings for $i \neq j$ will be published, while the encoding of s_i on the i-th chain will be kept private by user i. Therefore on the i-th chain only user i will be able to compute the session key. The exponents s_i are encoded in a "round robin" fashion; namely the i-th secret s_i is encoding on the chain of user j at edge $\ell = i + j - 1$, with index arithmetic modulo k.

We illustrate the protocol with $k = 2$ users. We have the following encodings:

$$\begin{aligned}\boldsymbol{A}_{1,1} \cdot \boldsymbol{D}_{1,1} &= s_1 \cdot \boldsymbol{A}_{1,2} + \boldsymbol{E}_{1,1} \pmod q \\ \boldsymbol{A}_{1,2} \cdot \boldsymbol{D}_{1,2} &= s_2 \cdot \boldsymbol{A}_0 + \boldsymbol{E}_{1,2} \pmod q \\ \boldsymbol{A}_{2,1} \cdot \boldsymbol{D}_{2,1} &= s_2 \cdot \boldsymbol{A}_{2,2} + \boldsymbol{E}_{2,1} \pmod q \\ \boldsymbol{A}_{2,2} \cdot \boldsymbol{D}_{2,2} &= s_1 \cdot \boldsymbol{A}_0 + \boldsymbol{E}_{2,2} \pmod q\end{aligned}$$

where $\boldsymbol{D}_{2,2}$ is public while $\boldsymbol{D}_{1,1}$ is kept private by User 1. Similarly $\boldsymbol{D}_{1,2}$ is public while $\boldsymbol{D}_{2,1}$ is kept private by User 2. Therefore User 1 can compute modulo q:

$$\begin{aligned}\boldsymbol{A}_{1,1} \cdot \boldsymbol{D}_{1,1} \cdot \boldsymbol{D}_{1,2} &= (s_1 \cdot \boldsymbol{A}_{1,2} + \boldsymbol{E}_{1,1}) \cdot \boldsymbol{D}_{1,2} \pmod q \\ &= s_1 \cdot s_2 \cdot \boldsymbol{A}_0 + s_1 \cdot \boldsymbol{E}_{1,2} + \boldsymbol{E}_{1,1} \cdot \boldsymbol{D}_{1,2} \pmod q\end{aligned}$$

and extract the most significant bits corresponding to $s_1 \cdot s_2 \cdot \boldsymbol{A}_0$. Similarly User 2 can compute modulo q:

$$\begin{aligned}\boldsymbol{A}_{2,1} \cdot \boldsymbol{D}_{2,1} \cdot \boldsymbol{D}_{2,2} &= (s_2 \cdot \boldsymbol{A}_{2,2} + \boldsymbol{E}_{2,1}) \cdot \boldsymbol{D}_{2,2} \pmod q \\ &= s_1 \cdot s_2 \cdot \boldsymbol{A}_0 + s_2 \cdot \boldsymbol{E}_{2,2} + \boldsymbol{E}_{2,1} \cdot \boldsymbol{D}_{2,2} \pmod q\end{aligned}$$

and extract the same most significant bits corresponding to $s_1 \cdot s_2 \cdot \boldsymbol{A}_0$.

The previous encodings $\boldsymbol{D}_{i,j}$ are generated by linear combination of public encodings, corresponding to secret exponents $t_{i,\ell}$ for $1 \leq \ell \leq N$, for large enough N. This means that we have the following public encodings:

$$\begin{aligned}
\boldsymbol{A}_{1,1} \cdot \boldsymbol{C}_{1,1,\ell} &= t_{1,\ell} \cdot \boldsymbol{A}_{1,2} + \boldsymbol{E}_{1,1,\ell} \pmod q \\
\boldsymbol{A}_{1,2} \cdot \boldsymbol{C}_{1,2,\ell} &= t_{2,\ell} \cdot \boldsymbol{A}_0 + \boldsymbol{E}_{1,2,\ell} \pmod q \\
\boldsymbol{A}_{2,1} \cdot \boldsymbol{C}_{2,1,\ell} &= t_{2,\ell} \cdot \boldsymbol{A}_{2,2} + \boldsymbol{E}_{2,1,\ell} \pmod q \\
\boldsymbol{A}_{2,2} \cdot \boldsymbol{C}_{2,2,\ell} &= t_{1,\ell} \cdot \boldsymbol{A}_0 + \boldsymbol{E}_{2,2,\ell} \pmod q
\end{aligned}$$

and the $\boldsymbol{D}_{i,j}$'s are generated by linear combination of the $\boldsymbol{C}_{i,j,\ell}$'s; more precisely, the pair $(\boldsymbol{D}_{1,1}, \boldsymbol{D}_{2,2})$ will be generated by User 1 by linear combination of the pairs $(\boldsymbol{C}_{1,1,\ell}, \boldsymbol{C}_{2,2,\ell})$, and similarly for User 2.

3 Cryptanalysis of GGH15

In the following we describe a cryptanalysis of the multipartite key-agreement protocol based on GGH15 multilinear maps.

3.1 Description with 2 Users

For simplicity we first consider the protocol with only 2 users, as in the previous section, using the same notations. Our attack proceeds in two steps.

1. In the first step, we are able to express one secret exponent s_1 as a linear combination of the other secret exponents $t_{1,\ell}$, using a variant of the Cheon *et al.* attack. However this does not immediately break the protocol, because the coefficients are not small.
2. In the second step, we compute an equivalent of the private encoding of User 1 by using the previous linear combination, while correcting the error due to the large coefficients.

First Step: Linear Relations In the first step of the attack, we show that we can express s_1 as a linear combinations of the $t_{1,\ell}$'s. We consider the public encodings $C_{i,j,\ell}$ and we let $C_{i,j}$ be one such set of encodings for a specific ℓ. We can compute the difference over R (not modulo q):

$$\begin{aligned}
\boldsymbol{\omega} &= A_{1,1} \cdot C_{1,1} \cdot C_{1,2} - A_{2,1} \cdot C_{2,1} \cdot C_{2,2} \\
&= t_1 \cdot E_{1,2} - E_{2,2} \cdot t_2 + E_{1,1} \cdot C_{1,2} - E_{2,1} \cdot C_{2,2}
\end{aligned} \tag{1}$$

We have that $\boldsymbol{\omega}$ is a vector of dimension m. Now an important step is to restrict ourselves to the first component of $\boldsymbol{\omega}$. Namely in order to apply the Cheon *et al.* attack, we would like to express $\boldsymbol{\omega}$ as the product of 2 vectors, where the left vector corresponds to User 1 and the right vector corresponds to User 2. Due to the "round-robin" fashion of exponent encodings, we should therefore swap the product $E_{2,1} \cdot C_{2,2}$, which we cannot do if we consider the full vector $\boldsymbol{\omega}$. Therefore we consider the first component ω', with:

$$\omega' = t_1 \cdot E'_{1,2} - E'_{2,2} \cdot t_2 + E_{1,1} \cdot C'_{1,2} - E_{2,1} \cdot C'_{2,2}$$

where both $C'_{1,2}$ and $C'_{2,2}$ are vectors of dimension m; similarly $E'_{1,2}$ and $E'_{2,2}$ are now scalars. We can now write:

$$\omega' = \begin{bmatrix} t_1 & E'_{2,2} & E_{1,1} & C'_{2,2} \end{bmatrix} \cdot \begin{bmatrix} E'_{1,2} \\ -t_2 \\ C'_{1,2} \\ -E_{2,1} \end{bmatrix}$$

Note that the two vectors in the product have dimension $2m+2$. As in the Cheon *et al.* attack, we can extend ω' to a matrix of dimension $(2m+2) \times (2m+2)$, but we take $2m+3$ rows instead of $2m+2$. This is done by considering $2m+3$ public encodings $C_{1,1,\ell}$ and $C_{2,2,\ell}$ corresponding to User 1, and similarly $2m+2$ encoding $C_{1,2,\ell}$ and $C_{2,1,\ell}$ corresponding to User 2. We obtain:

$$W = A \cdot B$$

where the matrix W has dimension $(2m+3) \times (2m+2)$, the matrix A has dimension $(2m+3) \times (2m+2)$, and the matrix B has dimension $(2m+2) \times (2m+2)$. We can find a vector u of dimension $(2m+3)$ such that $u \cdot W = 0$, which gives:

$$(u \cdot A) \cdot B = 0$$

With good probability the matrix B is full rank, which implies:

$$u \cdot A = 0$$

Such vector u gives a linear relation among the secret exponents $t_{1,i}$.

Moreover, if we assume now that both $D_{1,1}$ and $D_{2,2}$ are public (instead of only $D_{2,2}$ in the regular protocol), we can express s_1 as a linear combination of the $t_{1,i}$'s, over R. Namely we can compute $\boldsymbol{\omega}$ in (1) with the public $D_{1,1,}$ and $D_{2,2}$, instead of $C_{1,1}$ and $C_{2,2}$. This is done by collecting enough linear relations and taking gcd's to make sure that the coefficient of s_1 is 1. We obtain:

$$s_1 = \sum_j \alpha_j \cdot t_{1,j} \tag{2}$$

for some known α_j's. Note that we have the same linear relation for $E_{1,1}$:

$$E_{1,1} = \sum_j \alpha_j \cdot E_{1,1,j}$$

where by definition:

$$A_{1,1} \cdot D_{1,1} = s_1 \cdot A_{1,2} + E_{1,1} \pmod{q}$$

and also for the first component of $E_{2,2}$:

$$E'_{2,2} = \sum_j \alpha_j \cdot E'_{2,2,j}$$

We note that in the above attack we has used the private key $D_{1,1}$ of User 1 to derive the previous linear relations. Namely to derive such linear relations we need at least 2 encodings $D_{1,1}$ and $D_{2,2}$ corresponding to the same secret exponent s_1. Therefore when considering the key-agreement protocol for 2 users we don't really break the protocol. Our attack will be applicable for 3 users (or more), because for $k \geq 3$ users we will have $k-1 \geq 2$ *public* encodings $D_{i,i}$ corresponding to the same exponent s_1, namely for $2 \leq i \leq k$. We will use 2 such encodings in order to derive a linear relation for s_1 as in (2).

3.2 Second Step: Equivalent Private Key

In the second step, we will use the previous linear relation to derive an equivalent encoding for the private key $D_{1,1}$ of User 1. To illustrate our attack for 2 users, we artificially consider a 3rd chain:

$$A_{3,1} \cdot D_{3,1} = s_1 \cdot A_{3,2} + E_{3,1}$$
$$A_{3,2} \cdot D_{3,2} = s_2 \cdot A_0 + E_{3,2}$$

in which this time only $D_{3,2}$ is made public. Such 3rd chain will be available when considering the protocol with 3 users or more. We also have the public encodings $C_{3,1,\ell}$:

$$A_{3,1} \cdot C_{3,1,\ell} = t_{1,\ell} \cdot A_{3,2} + E_{3,1,\ell} \pmod{q} \tag{3}$$

which gives:

$$A_{3,1} \cdot C_{3,1,\ell} \cdot D_{3,2} = t_{1,\ell} \cdot s_2 \cdot A_0 + t_{1,\ell} \cdot E_{3,2} + E_{3,1,\ell} \cdot D_{3,2} \pmod{q}$$

In light of (2) it is natural to compute:

$$A_{3,1} \cdot \left(\sum_j \alpha_j \cdot C_{3,1,j} \right) \cdot D_{3,2} = s_1 \cdot s_2 \cdot A_0 + s_1 \cdot E_{3,2} + \left(\sum_j \alpha_j \cdot E_{3,1,j} \right) \cdot D_{3,2} \pmod{q}$$

The main problem is that the α_j's are not small so this does not reveal the high-order bits of $s_1 \cdot s_2 \cdot A_0$. In the following, we show how to derive an approximation of $\sum_j \alpha_j \cdot E_{3,1,j}$.

We consider some public encodings $C_{2,1}$ and $C_{3,2}$ corresponding to User 2, encoding the same exponent $t_{2,\ell}$. By taking the difference between Row 3 and Row 2, we can compute over R:

$$\begin{aligned} \boldsymbol{\Omega} &= \sum_j \alpha_j \cdot (A_{3,1} \cdot C_{3,1,j} \cdot C_{3,2} - A_{2,1} \cdot C_{2,1} \cdot C_{2,2,j}) \\ &= \sum_j \alpha_j \cdot (t_{1,j} \cdot E_{3,2} + E_{3,1,j} \cdot C_{3,2} - E_{2,2,j} \cdot t_2 - E_{2,1} \cdot C_{2,2,j}) \end{aligned}$$

As previously we restrict ourselves to the first component:

$$\begin{aligned} \Omega' &= \sum_j \alpha_j \cdot (t_{1,j} \cdot E'_{3,2} + E_{3,1,j} \cdot C'_{3,2} - E'_{2,2,j} \cdot t_2 - E_{2,1} \cdot C'_{2,2,j}) \\ &= s_1 \cdot E'_{3,2} - E'_{2,2} \cdot t_2 - E_{2,1} \cdot C'_{2,2} + \left(\sum_j \alpha_j \cdot E_{3,1,j} \right) \cdot C'_{3,2} \\ &= u + \left(\sum_j \alpha_j \cdot E_{3,1,j} \right) \cdot C'_{3,2} \end{aligned}$$

where u is small in R. We can now extend Ω' to a vector of dimension m, by using various encodings corresponding to $t_{2,\ell}$. We obtain a new vector $\boldsymbol{\Omega'}$ over R:

$$\boldsymbol{\Omega'} = \boldsymbol{u} + \left(\sum_j \alpha_j \cdot E_{3,1,j} \right) \cdot C'_{3,2}$$

where $C'_{3,2}$ is a $m \times m$ matrix with small coefficients.

Now the crucial observation is that because $\boldsymbol{u}$ is small we can get an approximation of $\sum_j \alpha_j \cdot E_{3,1,j}$ by reducing the vector $\boldsymbol{\Omega'}$ modulo $C'_{3,2}$. Therefore we obtain a vector E such that:

$$E' = \sum_j \alpha_j \cdot E_{3,1,j} - E$$

is small. Given the public encodings given by (3) we can therefore compute:

$$D'_{3,1} = \sum_j \alpha_j \cdot C_{3,1,j}$$

and we get:

$$A_{3,1} \cdot D'_{3,1} - E = s_1 \cdot A_{3,2} + E' \pmod q$$

for a small vector E'. Therefore we have obtained with $(D'_{3,1}, E)$ an equivalent of the private $D_{3,1}$, which breaks the protocol. More precisely, we can now compute:

$$\begin{aligned}(A_{3,1} \cdot D'_{3,1} - E) \cdot D_{3,2} &= (s_1 \cdot A_{3,2} + E') \cdot D_{3,2} \\ &= s_1 \cdot s_2 \cdot A_0 + s_1 \cdot E_{3,2} + E' \cdot D_{3,2}\end{aligned}$$

Since E' is small, this enables to extract the high-order bits of $s_1 \cdot s_2 \cdot A_0$ and breaks the protocol[1].

3.3 Cryptanalysis with 3 Users or More

We now consider the true cryptanalysis on 3 users (or more). For simplicity we restrict ourselves to 3 users; the attack works the same for more users. We have the following 3 chains:

$$\begin{array}{lllllll} A_{1,1} & t_1, C_{1,1} & A_{1,2} & t_2, C_{1,2} & A_{1,3} & t_3, C_{1,3} & A_0 \\ A_{2,1} & t_3, C_{2,1} & A_{2,2} & t_1, C_{2,2} & A_{2,3} & t_2, C_{2,3} & A_0 \\ A_{3,1} & t_2, C_{3,1} & A_{3,2} & t_3, C_{3,2} & A_{3,3} & t_1, C_{3,3} & A_0 \end{array}$$

For User 1, the encoding $D_{1,1}$ corresponding to s_1 is private, while $D_{2,2}$ and $D_{3,3}$ are public. Similarly for User 2, encodings $D_{1,2}$ and $D_{3,1}$ are public, and for user 3 encodings $D_{1,3}$ and $D_{2,1}$ are public.

First step: linear relations. In the first step, we will express s_1 as a linear combination of the $t_{1,i}$'s. For this we consider the rows 2 and 3, for which the encodings $D_{2,2}$ and $D_{3,3}$ of s_1 are public. We define the product encodings $C'_{2,2} = C_{2,1} \cdot C_{2,2}$ and $C'_{3,2} = C_{3,1} \cdot C_{3,2}$, and we have:

$$\begin{aligned} A_{2,1} \cdot C'_{2,2} &= t_1 \cdot t_3 \cdot A_{2,3} + E'_{2,2} \\ A_{2,3} \cdot C_{2,3} &= t_2 \cdot A_0 + E_{2,3} \\ A_{3,1} \cdot C'_{3,2} &= t_2 \cdot t_3 \cdot A_{3,3} + E'_{3,2} \\ A_{3,3} \cdot C_{3,3} &= t_1 \cdot A_0 + E_{3,3} \end{aligned}$$

for some small $E'_{2,2}$ and $E'_{3,2}$. As previously we can compute over R, restricting ourselves to the first component:

$$\begin{aligned} \omega &= A_{2,1} \cdot C'_{2,2} \cdot C_{2,3} - A_{3,1} \cdot C'_{3,2} \cdot C_{3,3} \\ &= t_1 \cdot t_3 \cdot E_{2,3} + E'_{2,2} \cdot C_{2,3} - t_2 \cdot t_3 \cdot E_{3,3} - E'_{3,2} \cdot C_{3,3} \\ &= \begin{bmatrix} t_1 & E'_{2,2} & E_{3,3} & C_{3,3} \end{bmatrix} \cdot \begin{bmatrix} t_3 \cdot E_{2,3} \\ C_{2,3} \\ -t_2 \cdot t_3 \\ -E'_{3,2} \end{bmatrix} \end{aligned}$$

As previously this enables to express s_1 as a linear combination of the $t_{1,\ell}$'s, as in (2).

[1] This is not a true attack since we have used the private encoding $D_{1,1}$ at Step 1. We only have a true attack for 3 users or more.

Second step: equivalent private-key. In this second step, we show how to compute an encoding equivalent to the private-key $D_{1,1}$. We now consider rows 1 and 3. We define the product encodings $C'_{1,2} = C_{1,2} \cdot C_{1,3}$ and $C'_{3,2} = C_{3,1} \cdot C_{3,2}$, with the following notations:

$$A_{1,2} \cdot C'_{1,2} = t_2 \cdot t_3 \cdot A_0 + E'_{1,2}$$
$$A_{3,1} \cdot C'_{3,2} = t_2 \cdot t_3 \cdot A_{3,3} + E'_{3,2}$$

We can then compute over R, using the coefficients α_j from the linear relation (2), restricting ourselves to the first component:

$$\begin{aligned}\Omega &= \sum_j \alpha_j \cdot (A_{1,1} \cdot C_{1,1,j} \cdot C'_{1,2} - A_{3,1} \cdot C'_{3,2} \cdot C_{3,3,j}) \\ &= \sum_j \alpha_j \cdot (t_{1,j} \cdot E'_{1,2} + E_{1,1,j} \cdot C'_{1,2} - t_2 \cdot t_3 \cdot E_{3,3,j} - E'_{3,2} \cdot C_{3,3,j}) \\ &= s_1 \cdot E'_{1,2} - t_2 \cdot t_3 \cdot E_{3,3} - E'_{3,2} \cdot C_{3,3} + \left(\sum_j \alpha_j \cdot E_{1,1,j} \right) \cdot C'_{1,2}\end{aligned}$$

As previously one can recover an approximation of $\sum_j \alpha_j \cdot E_{1,1,j}$, which enables to compute an equivalent $D'_{1,1}$ of the secret $D_{1,1}$, which breaks the key-agreement protocol.

More precisely, we can obtain a vector E such that:

$$E' = \sum_j \alpha_j \cdot E_{1,1,j} - E$$

is small. Then given the public encodings $C_{1,1,\ell}$ with:

$$A_{1,1} \cdot C_{1,1,\ell} = t_{1,\ell} \cdot A_{1,2} + E_{1,1,\ell} \pmod{q}$$

we can compute:

$$D'_{1,1} = \sum_j \alpha_j \cdot C_{1,1,j}$$

and we get:

$$A_{1,1} \cdot D'_{1,1} - E = s_1 \cdot A_{1,2} + E' \pmod{q}$$

for a small vector E'. Therefore we have obtained with $(D'_{1,1}, E)$ an equivalent of the private $D_{1,1}$, which breaks the protocol. Namely we can eventually compute from public parameters:

$$\begin{aligned}(A_{1,1} \cdot D'_{1,1} - E) \cdot D_{1,2} \cdot D_{1,3} &= (s_1 \cdot A_{1,2} + E') \cdot D_{1,2} \cdot D_{1,3} \\ &= s_1 \cdot s_2 \cdot s_3 \cdot A_0 + v\end{aligned}$$

for a small vector v. This enables to extract the high-order bits of $s_1 \cdot s_2 \cdot s_3 \cdot A_0$ and breaks the protocol.

References

[CHL+15] Jung Hee Cheon, Kyoohyung Han, Changmin Lee, Hansol Ryu, and Damien Stehlé. Cryptanalysis of the multilinear map over the integers. In *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I*, pages 3–12, 2015.

[GGH15] Craig Gentry, Sergey Gorbunov, and Shai Halevi. Graph-induced multilinear maps from lattices. In *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part II*, pages 498–527, 2015.