Cryptanalysis of GGH Map

Hu Yupu

Xidian University, China

2016.05

Background

Keywords

Multilinear maps, Multipartite key exchange (MKE), Witness encryption (WE), Lattice based cryptography.

Multilinear map

- ★ a leveled encoding system
  - ★ can multiply but cannot divide back
    - ★ goes further to extract limited information
      - ★ solution of a long-standing open problem
        - ★ a novel primitive which has many cryptographic applications

Background

Keywords

Multilinear maps, Multipartite key exchange (MKE), Witness encryption (WE), Lattice based cryptography.

Multilinear map

- ★ a leveled encoding system
  - ★ can multiply but cannot divide back
    - ★ goes further to extract limited information
      - ★ solution of a long-standing open problem
        - ★ a novel primitive which has many cryptographic applications

Background

Keywords

Multilinear maps, Multipartite key exchange (MKE), Witness encryption (WE), Lattice based cryptography.

Multilinear map

- ★ a leveled encoding system
  - ★ can multiply but cannot divide back
    - ★ goes further to extract limited information
      - ★ solution of a long-standing open problem
        - ★ a novel primitive which has many cryptographic applications

Background

Keywords

Multilinear maps, Multipartite key exchange (MKE), Witness encryption (WE), Lattice based cryptography.

Multilinear map

- ★ a leveled encoding system
  - ★ can multiply but cannot divide back
    - ★ goes further to extract limited information
      - ★ solution of a long-standing open problem
        - ★ a novel primitive which has many cryptographic applications

Background

Keywords

Multilinear maps, Multipartite key exchange (MKE), Witness encryption (WE), Lattice based cryptography.

Multilinear map

- ★ a leveled encoding system
- ★ can multiply but cannot divide back
- ★ goes further to extract limited information
- ★ solution of a long-standing open problem
- ★ a novel primitive which has many cryptographic applications

Background

Keywords

Multilinear maps, Multipartite key exchange (MKE), Witness encryption (WE), Lattice based cryptography.

Multilinear map

- ★ a leveled encoding system
  - ★ can multiply but cannot divide back
    - ★ goes further to extract limited information
      - ★ solution of a long-standing open problem
        - ★ a novel primitive which has many cryptographic applications

Background

GGH map

- the first candidate of K-linear maps for $K > 2$
  - from ideal lattice structure
    - a major candidate of multilinear maps
      - the best paper of EUROCRYPT 2013

Two classes of applications of GGH map

- applications with public tools for encoding
  - applications with hidden tools for encoding

Background

GGH map

- the first candidate of K-linear maps for $K > 2$
  - from ideal lattice structure
    - a major candidate of multilinear maps
      - the best paper of EUROCRYPT 2013

Two classes of applications of GGH map

- applications with public tools for encoding
  - applications with hidden tools for encoding

Background

GGH map

- the first candidate of K-linear maps for $K > 2$
  - from ideal lattice structure
    - a major candidate of multilinear maps
      - the best paper of EUROCRYPT 2013

Two classes of applications of GGH map

- applications with public tools for encoding
  - applications with hidden tools for encoding

Background

GGH map

- the first candidate of K-linear maps for $K > 2$
  - from ideal lattice structure
    - a major candidate of multilinear maps
      - the best paper of EUROCRYPT 2013

Two classes of applications of GGH map

- applications with public tools for encoding
  - applications with hidden tools for encoding

Background

GGH map

- the first candidate of K-linear maps for $K > 2$
  - from ideal lattice structure
    - a major candidate of multilinear maps
      - the best paper of EUROCRYPT 2013

Two classes of applications of GGH map

- applications with public tools for encoding
  - applications with hidden tools for encoding

Background

GGH map

- the first candidate of K-linear maps for $K > 2$
  - from ideal lattice structure
    - a major candidate of multilinear maps
      - the best paper of EUROCRYPT 2013

Two classes of applications of GGH map

- applications with public tools for encoding
  - applications with hidden tools for encoding

Background

GGH map

- the first candidate of K-linear maps for $K > 2$
  - from ideal lattice structure
    - a major candidate of multilinear maps
      - the best paper of EUROCRYPT 2013

Two classes of applications of GGH map

- applications with public tools for encoding
  - applications with hidden tools for encoding

Background

GGH map

- the first candidate of K-linear maps for $K > 2$
  - from ideal lattice structure
    - a major candidate of multilinear maps
      - the best paper of EUROCRYPT 2013

Two classes of applications of GGH map

- applications with public tools for encoding
  - applications with hidden tools for encoding

Background

In this paper, we show that applications of GGH map with public tools for encoding are not secure, and that one application of GGH map with hidden tools for encoding is not secure. On the basis of weak-DL attack presented by the authors themselves, we present several efficient attacks on GGH map, aiming at:

- multipartite key exchange (MKE)
- the instance of witness encryption (WE) based on the hardness of exact-3-cover ($X3C$) problem

A note

WE is another novel cryptographic primitive published on STOC 2013, and the instance of WE based on the hardness of $X3C$ problem is its first instance

Background

In this paper, we show that applications of GGH map with public tools for encoding are not secure, and that one application of GGH map with hidden tools for encoding is not secure. On the basis of weak-DL attack presented by the authors themselves, we present several efficient attacks on GGH map, aiming at:

- multipartite key exchange (MKE)
- the instance of witness encryption (WE) based on the hardness of exact-3-cover ($X3C$) problem

A note

WE is another novel cryptographic primitive published on STOC 2013, and the instance of WE based on the hardness of $X3C$ problem is its first instance

Background

In this paper, we show that applications of GGH map with public tools for encoding are not secure, and that one application of GGH map with hidden tools for encoding is not secure. On the basis of weak-DL attack presented by the authors themselves, we present several efficient attacks on GGH map, aiming at:

- multipartite key exchange (MKE)
- the instance of witness encryption (WE) based on the hardness of exact-3-cover $(X3C)$ problem

A note

WE is another novel cryptographic primitive published on STOC 2013, and the instance of WE based on the hardness of $X3C$ problem is its first instance

Background

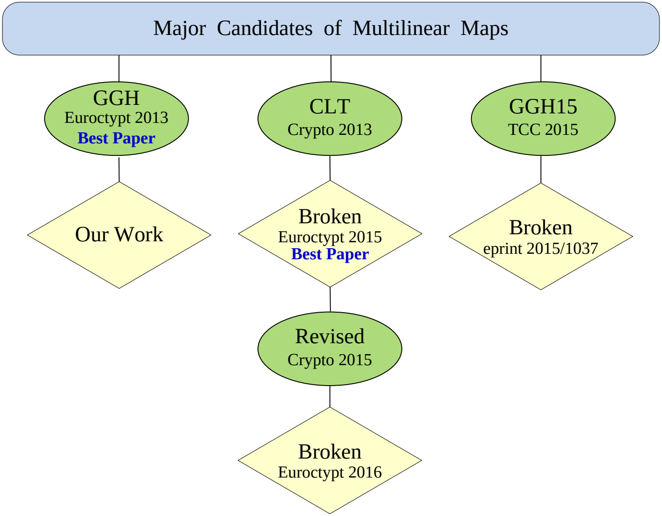
In this paper, we show that applications of GGH map with public tools for encoding are not secure, and that one application of GGH map with hidden tools for encoding is not secure. On the basis of weak-DL attack presented by the authors themselves, we present several efficient attacks on GGH map, aiming at:

- multipartite key exchange (MKE)
- the instance of witness encryption (WE) based on the hardness of exact-3-cover ($X3C$) problem

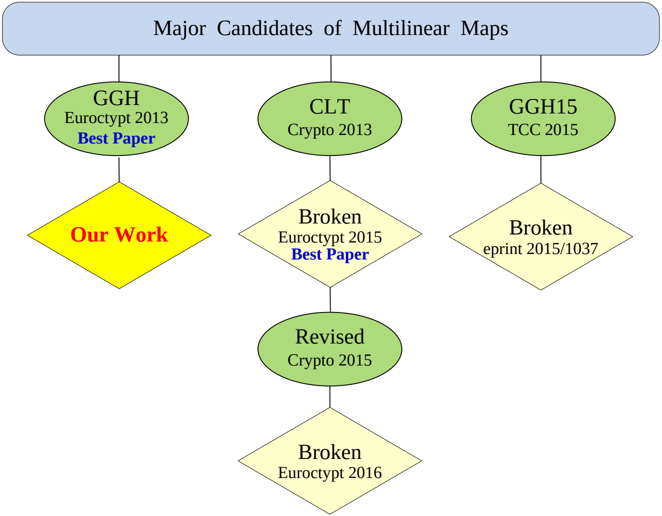
A note

WE is another novel cryptographic primitive published on STOC 2013, and the instance of WE based on the hardness of $X3C$ problem is its first instance

Background


Major Candidates of Multilinear Maps
GGH
Euroctypt 2013
Best Paper
CLT
Crypto 2013
GGH15
TCC 2015
Our Work
Broken
Euroctypt 2015
Best Paper
Broken
eprint 2015/1037
Revised
Crypto 2015
Broken
Euroctypt 2016

Background


Major Candidates of Multilinear Maps
GGH
Euroctypt 2013
Best Paper
CLT
Crypto 2013
GGH15
TCC 2015
Our Work
Broken
Euroctypt 2015
Best Paper
Broken
eprint 2015/1037
Revised
Crypto 2015
Broken
Euroctypt 2016

Contribution I

We use special modular operations, which we call modified encoding/zero-testing to drastically reduce the noise.

- Such reduction is enough to break MKE
- Moreover, such reduction negates K-GMDDH assumption, which is a basic security assumption

Notes

- *The procedure involves mostly simple algebraic manipulations, and rarely needs to use any lattice-reduction tools*
- *The key point is our special tools for modular operations*

Contribution I

We use special modular operations, which we call modified encoding/zero-testing to drastically reduce the noise.

- Such reduction is enough to break MKE
- Moreover, such reduction negates K-GMDDH assumption, which is a basic security assumption

Notes

- *The procedure involves mostly simple algebraic manipulations, and rarely needs to use any lattice-reduction tools*
- *The key point is our special tools for modular operations*

Contribution II

Under the condition of public tools for encoding, we break the instance of WE based on the hardness of $X3C$ problem.

Notes

- *To do so, we not only use modified encoding/zero-testing, but also introduce and solve "combined X3C problem", which is not difficult to solve*
- *In contrast with the assumption that multilinear map cannot be divided back, this attack includes a division operation, that is, solving an equivalent secret from a linear equation modular some principal ideal*
- *The quotient (the equivalent secret) is not small, so that modified encoding/zero-testing is needed to reduce size*
- *This attack is under an assumption that some two vectors are co-prime, which seems to be plausible*

Contribution II

Under the condition of public tools for encoding, we break the instance of WE based on the hardness of $X3C$ problem.

Notes

- *To do so, we not only use modified encoding/zero-testing, but also introduce and solve "combined X3C problem", which is not difficult to solve*
- *In contrast with the assumption that multilinear map cannot be divided back, this attack includes a division operation, that is, solving an equivalent secret from a linear equation modular some principal ideal*
- *The quotient (the equivalent secret) is not small, so that modified encoding/zero-testing is needed to reduce size*
- *This attack is under an assumption that some two vectors are co-prime, which seems to be plausible*

Contribution III

For hidden tools for encoding, we break the instance of WE based on the hardness of $X3C$ problem.

Notes

- *To do so, we construct level-2 encodings of 0, which are used as alternative tools for encoding*
- *Then, we break the scheme by applying modified encoding/zero-testing and combined $X3C$, where the modified encoding /zero-testing is an extended version*
- *This attack is under two assumptions, which seem to be plausible*

Contribution III

For hidden tools for encoding, we break the instance of WE based on the hardness of $X3C$ problem.

Notes

- *To do so, we construct level-2 encodings of 0, which are used as alternative tools for encoding*
- *Then, we break the scheme by applying modified encoding/zero-testing and combined $X3C$, where the modified encoding /zero-testing is an extended version*
- *This attack is under two assumptions, which seem to be plausible*

Contribution IV

- We check whether GGH structure can be simply revised to avoid our attack. We present cryptanalysis of two simple revisions of GGH map, aiming at MKE
- We show that MKE on these two revisions can be broken under the assumption that 2^K is polynomially large

Notes

- *To do so, we further extend our modified encoding/zero-testing*
- *These two simple revisions are "natural revisions", and cover "neighboring structures" of GGH structure*

Contribution IV

- We check whether GGH structure can be simply revised to avoid our attack. We present cryptanalysis of two simple revisions of GGH map, aiming at MKE
- We show that MKE on these two revisions can be broken under the assumption that 2^K is polynomially large

Notes

- *To do so, we further extend our modified encoding/zero-testing*
- *These two simple revisions are "natural revisions", and cover "neighboring structures" of GGH structure*

Main Technique I: Modified Encoding/zero-testing

Pre-procession: Weak-DL Attack

For the secret of each user, we have an equivalent secret which is the sum of original secret and a noise. These equivalent secrets cannot be encoded, because they are not small. We compute the product of these equivalent secrets, rather than computing their modular product.

Then our modified encoding/zero-testing is quite simple. It contains three simple operations, avoiding computing original secrets of users, and extracting same information. That is, it extracts same high-order bits of zero-tested message. The following table is a comparison between processing routines of GGH map and our work. It is a note of our claim that we can achieve the same purpose without knowing the secret of any user.

Main Technique I: Modified Encoding/zero-testing

Pre-procession: Weak-DL Attack

For the secret of each user, we have an equivalent secret which is the sum of original secret and a noise. These equivalent secrets cannot be encoded, because they are not small. We compute the product of these equivalent secrets, rather than computing their modular product.

Then our modified encoding/zero-testing is quite simple. It contains three simple operations, avoiding computing original secrets of users, and extracting same information. That is, it extracts same high-order bits of zero-tested message. The following table is a comparison between processing routines of GGH map and our work. It is a note of our claim that we can achieve the same purpose without knowing the secret of any user.

Main Technique I: Modified Encoding/zero-testing

Table: Processing routines

GGH map	secrets → encodings → **modular** product → zero-testing → high-order bits
Our work	**equivalent** secrets → product → **modified** encoding/zero-testing → high-order bits

Main Technique I: Modified Encoding/zero-testing

Table: Processing routines

GGH map	secrets $\longrightarrow$ encodings $\longrightarrow$ **modular** product $\longrightarrow$ zero-testing $\longrightarrow$ high-order bits
Our work	**equivalent** secrets $\longrightarrow$ product $\longrightarrow$ **modified** encoding/zero-testing $\longrightarrow$ high-order bits

Main Technique I: Modified Encoding/zero-testing

Table: Processing routines

GGH map	secrets → encodings → **modular** product → zero-testing → high-order bits
Our work	**equivalent** secrets → product → **modified** encoding/zero-testing → high-order bits

Main Technique II: Solving Combined $X3C$ Problem

The reason that $X3C$ problem can be transformed into a combined $X3C$ problem is that the special structure of GGH map sometimes makes division possible.

We can solve combined $X3C$ problem with non-negligible probability and break the instance of WE based on the hardness of $X3C$ problem for public tools of encoding.

Main Technique II: Solving Combined $X3C$ Problem

The reason that $X3C$ problem can be transformed into a combined $X3C$ problem is that the special structure of GGH map sometimes makes division possible.

$$\boxed{A \times B = C}$$

We can solve combined $X3C$ problem with non-negligible probability and break the instance of WE based on the hardness of $X3C$ problem for public tools of encoding.

Main Technique II: Solving Combined $X3C$ Problem

The reason that $X3C$ problem can be transformed into a combined $X3C$ problem is that the special structure of GGH map sometimes makes division possible.

$$\boxed{A \times B = C \quad \Longrightarrow \quad B = A^{-1} \times C}$$

We can solve combined $X3C$ problem with non-negligible probability and break the instance of WE based on the hardness of $X3C$ problem for public tools of encoding.

Main Technique III: Finding Alternative Encoding Tools

When encoding tools are hidden, we can use redundant information to construct alternative encoding tools. For example, there are many redundant pieces beside $X3C$. Encodings of these redundant pieces can be composed into several level-2 encodings of 0.

Only one level-2 encoding of 0 is enough to break the instance of WE based on the hardness of $X3C$ problem for hidden tools of encoding. This technique can be adapted to other applications of GGH map, where although encoding tools are hidden, a large number of redundant information are needed to protect some secrets.

Main Technique III: Finding Alternative Encoding Tools

When encoding tools are hidden, we can use redundant information to construct alternative encoding tools. For example, there are many redundant pieces beside $X3C$. Encodings of these redundant pieces can be composed into several level-2 encodings of 0.

Only one level-2 encoding of 0 is enough to break the instance of WE based on the hardness of $X3C$ problem for hidden tools of encoding. This technique can be adapted to other applications of GGH map, where although encoding tools are hidden, a large number of redundant information are needed to protect some secrets.

Main Technique III: Finding Alternative Encoding Tools

When encoding tools are hidden, we can use redundant information to construct alternative encoding tools. For example, there are many redundant pieces beside $X3C$. Encodings of these redundant pieces can be composed into several level-2 encodings of 0.

Only one level-2 encoding of 0 is enough to break the instance of WE based on the hardness of $X3C$ problem for hidden tools of encoding. This technique can be adapted to other applications of GGH map, where although encoding tools are hidden, a large number of redundant information are needed to protect some secrets.

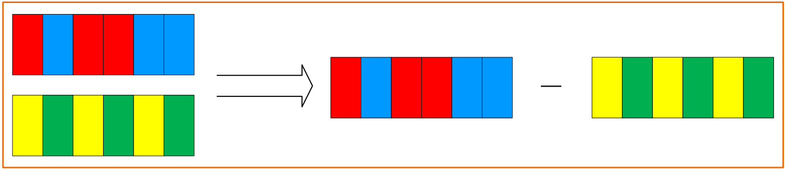
Main Technique III: Finding Alternative Encoding Tools

When encoding tools are hidden, we can use redundant information to construct alternative encoding tools. For example, there are many redundant pieces beside $X3C$. Encodings of these redundant pieces can be composed into several level-2 encodings of 0.

Only one level-2 encoding of 0 is enough to break the instance of WE based on the hardness of $X3C$ problem for hidden tools of encoding. This technique can be adapted to other applications of GGH map, where although encoding tools are hidden, a large number of redundant information are needed to protect some secrets.

Main Technique III: Finding Alternative Encoding Tools

When encoding tools are hidden, we can use redundant information to construct alternative encoding tools. For example, there are many redundant pieces beside $X3C$. Encodings of these redundant pieces can be composed into several level-2 encodings of 0.



Only one level-2 encoding of 0 is enough to break the instance of WE based on the hardness of $X3C$ problem for hidden tools of encoding. This technique can be adapted to other applications of GGH map, where although encoding tools are hidden, a large number of redundant information are needed to protect some secrets.

Parameter Setting

- $\mathbb{Q}$: rational numbers
- $\mathbb{Z}$: integers
- $\mathbb{Q}^n$ and $\mathbb{Z}^n$: n-dimensional row vectors
- $R = \mathbb{Z}[X]/(X^n + 1)$: polynomial ring
- $R_q = R/qR = \mathbb{Z}_q[X]/(X^n + 1)$ for a (large enough) integer q
- "mod q" is redefined
- $\{g, z, a, b^{(1)}, b^{(2)}, h\} \subseteq R$ are kept from all users
  - $\{g, a, b^{(1)}, b^{(2)}\}$ are small
  - h is somewhat small
  - z is random
  - We use principal ideal $\langle g \rangle$

Parameter Setting

- $\mathbb{Q}$: rational numbers
- $\mathbb{Z}$: integers
- $\mathbb{Q}^n$ and $\mathbb{Z}^n$: n-dimensional row vectors
- $R = \mathbb{Z}[X]/(X^n + 1)$: polynomial ring
- $R_q = R/qR = \mathbb{Z}_q[X]/(X^n + 1)$ for a (large enough) integer q
- "mod q" is redefined
- $\{g, z, a, b^{(1)}, b^{(2)}, h\} \subseteq R$ are kept from all users
  - $\{g, a, b^{(1)}, b^{(2)}\}$ are small
  - h is somewhat small
  - z is random
  - We use principal ideal $\langle g \rangle$

Parameter Setting

- y (level-1 encoding of 1): $y = (1 + ag)z^{-1} (\text{mod } q)$
- $\{x^{(i)}, i = 1, 2\}$ (level-1 encoding of 0): $x^{(i)} = b^{(i)} g z^{-1} (\text{mod } q)$, $i = 1, 2$
- p_{zt} (level-K zero-testing parameter): $p_{zt} = (hz^K g^{-1})(\text{mod } q)$

- p_{zt} is always public
- y and $\{x^{(i)}, i = 1, 2\}$ are called tools for encoding
- For MKE, y and $\{x^{(i)}, i = 1, 2\}$ are public
- For WE, y and $\{x^{(i)}, i = 1, 2\}$ can be either public or hidden

Parameter Setting

- y (level-1 encoding of 1): $y = (1 + ag)z^{-1} (\text{mod } q)$
- $\{x^{(i)}, i = 1, 2\}$ (level-1 encoding of 0): $x^{(i)} = b^{(i)} g z^{-1} (\text{mod } q)$, $i = 1, 2$
- p_{zt} (level-K zero-testing parameter): $p_{zt} = (hz^K g^{-1}) (\text{mod } q)$

- p_{zt} is always public
- y and $\{x^{(i)},\ i = 1, 2\}$ are called tools for encoding
- For MKE, y and $\{x^{(i)},\ i = 1, 2\}$ are public
- For WE, y and $\{x^{(i)},\ i = 1, 2\}$ can be either public or hidden

GGH Map

Suppose a user has a secret $v \in R$, which is a short element.

(1) He encodes v into V

- *He secretly samples short elements* $\{u^{(i)} \in R, i = 1, 2\}$. *He computes noised encoding* $V = vy + (u^{(1)}x^{(1)} + u^{(2)}x^{(2)})$*(mod* q*), where* vy*(mod* q*) and* $(u^{(1)}x^{(1)} + u^{(2)}x^{(2)})$*(mod* q*) are respectively encoded secret and encoded noise*

(2) 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.

GGH Map

Suppose a user has a secret $v \in R$, which is a short element.

(1) He encodes v into V

- *He secretly samples short elements* $\{u^{(i)} \in R, i = 1, 2\}$. *He computes noised encoding* $V = vy + (u^{(1)}x^{(1)} + u^{(2)}x^{(2)})$(*mod* q), *where* vy(*mod* q) *and* $(u^{(1)}x^{(1)} + u^{(2)}x^{(2)})$(*mod* q) *are respectively encoded secret and encoded noise*

(2) 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.

Application 1: MKE

Suppose that $K+1$ users want to generate KEY, a common shared key, by public discussion.

To do so, each user k_0 uses his secret $v^{(k_0)}$ and other users' encodings $\{V^{(k)}, k \neq k_0\}$, to compute the modular product

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

Then KEY is its high-order bits, with no relation to k_0.

Application 1: MKE

Suppose that $K+1$ users want to generate KEY, a common shared key, by public discussion.

To do so, each user k_0 uses his secret $v^{(k_0)}$ and other users' encodings $\{V^{(k)}, k \neq k_0\}$, to compute the modular product

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

Then KEY is its high-order bits, with no relation to k_0.

$X3C$ Problem [3, 24]

- a piece: a subset of $\{1, 2, \ldots, 3K\}$ containing 3 integers
- $X3C$: a collection of K pieces without intersection
- the $X3C$ problem: for arbitrarily given $N(K)$ different pieces with an $X3C$, find it

A note

Intuitively, the $X3C$ problem is often not hard when $N(K) \leq O(K)$, because $X3C$ is not hidden well. An extreme example is that if the number i is contained by only one piece $\{i, j, k\}$, then $\{i, j, k\}$ is certainly from $X3C$. Picking up $\{i, j, k\}$ and abandoning those pieces containing j or k, then other pieces form a reduced $X3C$ problem on $\{1, 2, \ldots, 3K\} - \{i, j, k\}$. So that $N(K) \geq O(K^2)$ to avoid weak case.

Encryption

The encrypter generates $EKEY$ as follows. He

(1) samples short elements $v^{(1)}, v^{(2)}, \cdots, v^{(3K)} \in R$

(2) computes $v^{(1)}v^{(2)} \cdots v^{(3K)} y^K \ p_{zt} (\text{mod } q)$

(3) takes $EKEY$ as its high-order bits

- *He can use $EKEY$ as the key to encrypt any plaintext.*

Then he hides $EKEY$ into pieces as follows. He

(1) randomly generates $N(K)$ different pieces of $\{1, 2, \cdots, 3K\}$, with an $X3C$

(2) for each piece $\{i_1, i_2, i_3\}$, encodes the product $v^{(i_1)}v^{(i_2)}v^{(i_3)}$ into $V^{\{i_1, i_2, i_3\}}$

(3) publishes all $V^{\{i_1, i_2, i_3\}}$

Encryption

The encrypter generates $EKEY$ as follows. He

(1) samples short elements $v^{(1)}, v^{(2)}, \cdots, v^{(3K)} \in R$

(2) computes $v^{(1)}v^{(2)} \cdots v^{(3K)} y^K \; p_{zt} (\text{mod } q)$

(3) takes $EKEY$ as its high-order bits

- *He can use $EKEY$ as the key to encrypt any plaintext.*

Then he hides $EKEY$ into pieces as follows. He

(1) randomly generates $N(K)$ different pieces of $\{1, 2, \cdots, 3K\}$, with an $X3C$

(2) for each piece $\{i_1, i_2, i_3\}$, encodes the product $v^{(i_1)}v^{(i_2)}v^{(i_3)}$ into $V^{\{i_1,i_2,i_3\}}$

(3) publishes all $V^{\{i_1,i_2,i_3\}}$

Decryption

The one who knows $X3C$ computes the zero-test of

$$\prod_{\{i_1,i_2,i_3\}\in X3C} V^{\{i_1,i_2,i_3\}} (\text{mod } q).$$

That is, he/she computes

$$p_{zt} \prod_{\{i_1,i_2,i_3\}\in X3C} V^{\{i_1,i_2,i_3\}} (\text{mod } q).$$

Then, $EKEY$ is its high-order bits.

A note

In other words, $p_{zt} \prod_{\{i_1,i_2,i_3\}\in X3C} V^{\{i_1,i_2,i_3\}} (\text{mod } q)$ is the modular sum of two terms, the first term is zero-tested message $v^{(1)} v^{(2)} \ldots v^{(3K)} (1+ag)^K hg^{-1} \pmod q$, while the second term is zero-tested noise which doesn't affect high-order bits of $p_{zt} \prod_{\{i_1,i_2,i_3\}\in X3C} V^{\{i_1,i_2,i_3\}} (\text{mod } q)$.

Decryption

The one who knows $X3C$ computes the zero-test of

$$\prod_{\{i_1,i_2,i_3\}\in X3C} V^{\{i_1,i_2,i_3\}} (\text{mod } q).$$

That is, he/she computes

$$p_{zt} \prod_{\{i_1,i_2,i_3\}\in X3C} V^{\{i_1,i_2,i_3\}} (\text{mod } q).$$

Then, $EKEY$ is its high-order bits.

A note

In other words, $p_{zt} \prod_{\{i_1,i_2,i_3\}\in X3C} V^{\{i_1,i_2,i_3\}}$ (mod q) is the modular sum of two terms, the first term is zero-tested message $v^{(1)}v^{(2)}\ldots v^{(3K)}(1+ag)^K hg^{-1}$ (mod q), while the second term is zero-tested noise which doesn't affect high-order bits of $p_{zt} \prod_{\{i_1,i_2,i_3\}\in X3C} V^{\{i_1,i_2,i_3\}}$ (mod q).

Our Special Tools

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

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

Modified Encoding/Zero-testing

By GGH's weak-DL attack, we can get equivalent secrets $v^{(0,k)}$ of each user's secret $v^{(k)}$ for $k = 1, \ldots, K+1$ such that $v^{(0,k)} \equiv v^{(k)} \pmod{\langle g \rangle}$.

Now we transform $\prod_{k=1}^{K+1} v^{(0,k)}$ by our modified encoding/zero-testing. 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 $\xi \in R$.

Modified Encoding/Zero-testing

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.

Modified Encoding/Zero-testing

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.

Modified Encoding/Zero-testing

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. MKE has been broken. More important is that K-GMDDH assumption (Assumption 5.1 of [2]) is negated.

Thank you !

References I

[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)

References II

[6] Bellare, M., Hoang, V.T.: Adaptive Witness Encryption and Asymmetric Password-Based Cryptography. Cryptology ePrint Archive, Report 2013/704 (2013)

[7] 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)

[8] Garg, S., Gentry, C., Halevi, S., Raykova, M., Sahai, A., Waters, B.: Candidate Indistinguishability Obfuscation and Functional Encryption for all Circuits. In: FOCS (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)

References III

[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., Stehlé, D., Steinfeld, R.: GGHLite: More 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., Stehlé. D: Cryptanalysis of the Multilinear Map over the Integers. In: Oswald, E., Fischlin, M. (ed.) EUROCRYPT 2015, Part I. LNCS, vol. 9056, pp. 3–12. Springer, Heidelberg (2015)

References IV

[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] Gentry, C., Gorbunov, S., Halevi, S.: Graph-Induced Multilinear Maps from Lattices. In: Dodis, Y. and Nielsen, J.B. (ed.) TCC 2015, Part II, LNCS, vol. 9015, pp. 498–527. Springer, Heidelberg (2015)

[19] Coron, J.-S., Gentry, C., Halevi, S., Lepoint, T., Maji H.K., Miles, E., Raykova, M., Sahai, A., Tibouchi, M.: Zeroizing Without Low-level Zeroes: New MMAP Attacks and their Limitations. In: Gennaro, R., Robshaw, M. (ed.) CRYPTO 2015, Part I. LNCS, vol. 9215, pp. 247–266. Springer, Heidelberg (2015)

References V

[20] Coron, J.-S., Lepoint, T., Tibouchi, M.: New Multilinear Maps over the Integers. In: Gennaro, R., Robshaw, M. (ed.) CRYPTO 2015, Part I. LNCS, vol. 9215, pp. 267–286. Springer, Heidelberg (2015)

[21] Cheon, J.H., Han, K., Lee, C., Ryu, H.: Cryptanalysis of the New CLT Multilinear Maps. Cryptology ePrint Archive, Report 2015/934 (2015)

[22] Minaud, B., Fouque, P.-A.: Cryptanalysis of the New Multilinear Map over the Integers. Cryptology ePrint Archive, Report 2015/941 (2015)

[23] Babai, L.: On Lovász' lattice reduction and the nearest lattice point problem. Combinatorica 6(1): 1–13 (1986)

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

[25] Gu, C.: Multilinear Maps Using Ideal Lattices without Encodings of Zero. Cryptology ePrint Archive, Report 2015/023 (2015)

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

References VI

[27] Albrecht, M.R.: Sage Code for GGH Cryptanalysis by Hu and Jia. Available at https://martinralbrecht.wordpress.com/2015/04/13/sage-code-for-ggh-cryptanalysis-by-hu-and-jia/