RSAConference2019

San Francisco | March 4–8 | Moscone Center

BETTER.

SESSION ID: CRYP-W02

Tight Reductions for Diffie-Hellman Variants in the Algebraic Group Model

Taiga Mizuide[1], Atsushi Takayasu[1,2], Tsuyoshi Takagi[1]

[1]The University of Tokyo, Japan

[2]National Institute of Advanced Industrial Science and Technology, Japan

#RSAC

Summary of Our Result

- In this talk, we show tight equivalences between the DL and several Diffie-Hellman variants in the *algebraic group model* defined in [FKL@Crypto'18].
- The results imply information theoretic lower bounds for solving the Diffie-Hellman variants in the generic group model.
- The most advantage is that we obtain the results through *very very very simple techniques*.

Discrete Logarithm Problem

- Discrete Logarithm (DL) Problem
  Input: $(\mathbb{G}, g, p)$ and $X = g^x; x \leftarrow \mathbb{Z}_p$
  Solution: x

Discrete Logarithm Problem

- Discrete Logarithm (DL) Problem
  Input: $(\mathbb{G}, g, p)$ and $X = g^x; x \leftarrow \mathbb{Z}_p$
  Solution: x

- Numerous cryptographic protocols have been proposed over cyclic groups $\mathbb{G}$ by assuming that the DL problem is computationally infeasible in the groups (e.g., elliptic curves).

Discrete Logarithm Problem

- Discrete Logarithm (DL) Problem
  Input: $(\mathbb{G}, g, p)$ and $X = g^x; x \leftarrow \mathbb{Z}_p$
  Solution: x

- Numerous cryptographic protocols have been proposed over cyclic groups $\mathbb{G}$ by assuming that the DL problem is computationally infeasible in the groups (e.g., elliptic curves).

- A number of DL algorithms have been proposed;
  - sub-exponential algorithms (e.g., NFS) working in specific groups,
  - $O(\sqrt{p})$ time algorithms working in any cyclic groups.

Discrete Logarithm Problem

- Discrete Logarithm (DL) Problem
  Input: $(\mathbb{G}, g, p)$ and $X = g^x; x \leftarrow \mathbb{Z}_p$
  Solution: x

- Numerous cryptographic protocols have been proposed over cyclic groups $\mathbb{G}$ by assuming that the DL problem is computationally infeasible in the groups (e.g., elliptic curves).

- A number of DL algorithms have been proposed;
  - sub-exponential algorithms (e.g., NFS) working in specific groups,
  - $O(\sqrt{p})$ time algorithms working in any cyclic groups.

- The latter is called *generic algorithms*, where $O(\sqrt{p})$ times group operations is proved to be an information theoretic lower bound [Shoup@EC'97].

Diffie-Hellman Problem and Its Variants

- Computational Diffie-Hellman (CDH) Problem
  Input: $(\mathbb{G}, g, p)$ and $(X_1 = g^{x_1}, X_2 = g^{x_2})$; $(x_1, x_2) \leftarrow \mathbb{Z}_p^2$
  Solution: $g^{x_1 x_2}$

Diffie-Hellman Problem and Its Variants

- Computational Diffie-Hellman (CDH) Problem [DH@IEEE TIT'76]
  Input: $(\mathbb{G}, g, p)$ and $(X_1 = g^{x_1}, X_2 = g^{x_2})$; $(x_1, x_2) \leftarrow \mathbb{Z}_p^2$
  Solution: $g^{x_1 x_2}$

- *k*-Exponent Diffie-Hellman (*k*-EDH) Problem [MW@Crypto'96],[BDS@AC'98]
  Input: $(\mathbb{G}, g, p)$ and $X = g^x$; $x \leftarrow \mathbb{Z}_p$
  Solution: g^{x^k}

- *k*-Party Diffie-Hellman (*k*-PDH) Problem [Bis@IET Information Security'08]
  Input: $(\mathbb{G}, g, p)$ and $(X_1 = g^{x_1}, \ldots, X_k = g^{x_k})$; $(x_1, \ldots, x_k) \leftarrow \mathbb{Z}_p^k$
  Solution: $g^{x_1 \cdots x_k}$

- and so on

Diffie-Hellman Problem and Its Variants

- Computational Diffie-Hellman (CDH) Problem [DH@IEEE TIT'76]
  Input: $(\mathbb{G}, g, p)$ and $(X_1 = g^{x_1}, X_2 = g^{x_2})$; $(x_1, x_2) \leftarrow \mathbb{Z}_p^2$
  Solution: $g^{x_1 x_2}$

- *k*-Exponent Diffie-Hellman (*k*-EDH) Problem [MW@Crypto'96],[BDS@AC'98]
  Input: $(\mathbb{G}, g, p)$ and $X = g^x$; $x \leftarrow \mathbb{Z}_p$
  Solution: g^{x^k}

- *k*-Party Diffie-Hellman (*k*-PDH) Problem [Bis@IET Information Security'08]
  Input: $(\mathbb{G}, g, p)$ and $(X_1 = g^{x_1}, \ldots, X_k = g^{x_k})$; $(x_1, \ldots, x_k) \leftarrow \mathbb{Z}_p^k$
  Solution: $g^{x_1 \cdots x_k}$

- and so on

✓ We should try to study computational complexities of these problems. If possible, we want to make computational reductions from the DL to these problems although it seems infeasible in the standard computational models…

Generic Group Model (GGM)

- Generic Group Model (GGM) [Shoup@EC’97]
  Restricted computational model, where a generic adversary
  - is *not* able to exploit group specific structures,
  - is able to receive group elements *only* via abstract handles.

Generic Group Model (GGM)

- Generic Group Model (GGM) [Shoup@EC'97]
  Restricted computational model, where a generic adversary
  - is *not* able to exploit group specific structures,
  - is able to receive group elements *only* via abstract handles.

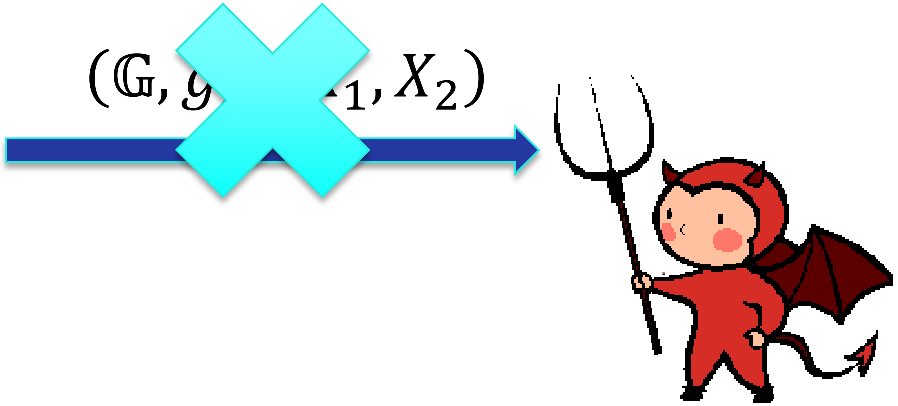
$(\mathbb{G}, g, p) \leftarrow \mathcal{G}(1^\lambda)$ and
$(X_i = g^{x_i}); (x_1, x_2) \leftarrow \mathbb{Z}_p^2$

$\xrightarrow{(\mathbb{G}, g, p, X_1, X_2)}$

Generic Group Model (GGM)

- Generic Group Model (GGM) [Shoup@EC'97]
  Restricted computational model, where a generic adversary
  - is *not* able to exploit group specific structures,
  - is able to receive group elements *only* via abstract handles.

$(\mathbb{G}, g, p) \leftarrow \mathcal{G}(1^\lambda)$ and
$(X_i = g^{x_i}); (x_1, x_2) \leftarrow \mathbb{Z}_p^2$



Generic Group Model (GGM)

- Generic Group Model (GGM) [Shoup@EC'97]
  Restricted computational model, where a generic adversary
  - is *not* able to exploit group specific structures,
  - is able to receive group elements *only* via abstract handles.

$(\mathbb{G}, g, p) \leftarrow \mathcal{G}(1^\lambda)$ and
$(X_i = g^{x_i}); (x_1, x_2) \leftarrow \mathbb{Z}_p^2$
$\mathit{O0} \leftarrow g$
$\mathit{O1} \leftarrow g^{x_1}$
$\mathit{O2} \leftarrow g^{x_2}$

$\xrightarrow{\cancel{(\mathbb{G}, g, X_1, X_2)}}$

$\xrightarrow{(\mathbb{G}, p, \mathit{O0}, \mathit{O1}, \mathit{O2})}$

Generic Group Model (GGM)

- Generic Group Model (GGM) [Shoup@EC'97]
  Restricted computational model, where a generic adversary
  - is *not* able to exploit group specific structures,
  - is able to receive group elements *only* via abstract handles.

$(\mathbb{G}, g, p) \leftarrow \mathcal{G}(1^\lambda)$ and
$(X_i = g^{x_i}); (x_1, x_2) \leftarrow \mathbb{Z}_p^2$
$00 \leftarrow g$
$01 \leftarrow g^{x_1}$
$02 \leftarrow g^{x_2}$

$\cancel{(\mathbb{G}, g, X_1, X_2)}$

$(\mathbb{G}, p, 00, 01, 02)$

$\mathcal{O}$

Generic Group Model (GGM)

- Generic Group Model (GGM) [Shoup@EC’97]
  Restricted computational model, where a generic adversary
  - is *not* able to exploit group specific structures,
  - is able to receive group elements *only* via abstract handles.

$(\mathbb{G}, g, p) \leftarrow \mathcal{G}(1^\lambda)$ and
$(X_i = g^{x_i}); (x_1, x_2) \leftarrow \mathbb{Z}_p^2$
$00 \leftarrow g$
$01 \leftarrow g^{x_1}$
$02 \leftarrow g^{x_2}$

$(\mathbb{G}, g, X_1, X_2)$

$(\mathbb{G}, p, 00, 01, 02)$

$(01, 02)$

04

$\mathcal{O}$

$g^{x_1} \cdot g^{x_2} = g^{x_1+x_2}$
$04 \leftarrow g^{x_1+x_2}$

Generic Group Model (GGM)

- Generic Group Model (GGM) [Shoup@EC'97]
  Restricted computational model, where a generic adversary
  - is *not* able to exploit group specific structures,
  - is able to receive group elements *only* via abstract handles.

$(\mathbb{G}, g, p) \leftarrow \mathcal{G}(1^\lambda)$ and
$(X_i = g^{x_i}); (x_1, x_2) \leftarrow \mathbb{Z}_p^2$
$\mathcal{O}0 \leftarrow g$
$\mathcal{O}1 \leftarrow g^{x_1}$
$\mathcal{O}2 \leftarrow g^{x_2}$

$\cancel{(\mathbb{G}, g, \ldots, X_2)}$

$(\mathbb{G}, p, \mathcal{O}0, \mathcal{O}1, \mathcal{O}2)$

$(\mathcal{O}1, \mathcal{O}2)$

$\mathcal{O}4$

$(\mathcal{O}1, a \in \mathbb{Z}_p)$

$\mathcal{O}5$

$\mathcal{O}$

$g^{x_1} \cdot g^{x_2} = g^{x_1+x_2}$
$\mathcal{O}4 \leftarrow g^{x_1+x_2}$

$(g^{x_1})^a = g^{a x_1}$
$\mathcal{O}5 \leftarrow g^{a x_1}$

Generic Group Model (GGM)

- Generic Group Model (GGM) [Shoup@EC'97]
  Restricted computational model, where a generic adversary
  - is *not* able to exploit group specific structures,
  - is able to receive group elements *only* via abstract handles.

$(\mathbb{G}, g, p) \leftarrow \mathcal{G}(1^\lambda)$ and
$(X_i = g^{x_i})$; $(x_1, x_2) \leftarrow \mathbb{Z}_p^2$
$00 \leftarrow g$
$01 \leftarrow g^{x_1}$
$02 \leftarrow g^{x_2}$

$\cancel{(\mathbb{G}, g, X_1, X_2)} \rightarrow$

$(\mathbb{G}, p, 00, 01, 02) \rightarrow$

$\leftarrow \mathcal{XX}$

$(01, 02) \rightarrow$

$\leftarrow 04$

$(01, a \in \mathbb{Z}_p) \rightarrow$

$\leftarrow 05$

$\mathcal{O}$

$g^{x_1} \cdot g^{x_2} = g^{x_1+x_2}$
$04 \leftarrow g^{x_1+x_2}$
$(g^{x_1})^a = g^{a x_1}$
$05 \leftarrow g^{a x_1}$

Generic Group Model (GGM)

- Generic Group Model (GGM) [Shoup@EC'97]
  Restricted computational model, where a generic adversary
  - is *not* able to exploit group specific structures,
  - is able to receive group elements *only* via abstract handles.

Generic Group Model (GGM)

- Generic Group Model (GGM) [Shoup@EC'97]
  Restricted computational model, where a generic adversary
  - is *not* able to exploit group specific structures,
  - is able to receive group elements *only* via abstract handles.
- Advantage of GGM
  - Generic algorithms work in any groups.
  - Information theoretic lower bounds for computational problems (e.g., $O(\sqrt{p})$ for the DL).

Generic Group Model (GGM)

- Generic Group Model (GGM) [Shoup@EC'97]
  Restricted computational model, where a generic adversary
  - is *not* able to exploit group specific structures,
  - is able to receive group elements *only* via abstract handles.
- Advantage of GGM
  - Generic algorithms work in any groups.
  - Information theoretic lower bounds for computational problems (e.g., $O(\sqrt{p})$ for the DL).
- Disadvantage of GGM
  - There may be faster algorithms in concrete groups (e.g., sub-exponential DL algorithms).

Generic Group Model (GGM)

- Generic Group Model (GGM) [Shoup@EC'97]
  Restricted computational model, where a generic adversary
  - is *not* able to exploit group specific structures,
  - is able to receive group elements *only* via abstract handles.
- Advantage of GGM
  - Generic algorithms work in any groups.
  - Information theoretic lower bounds for computational problems (e.g., $O(\sqrt{p})$ for the DL).
- Disadvantage of GGM
  - There may be faster algorithms in concrete groups (e.g., sub-exponential DL algorithms).

Can we obtain similar results in
less restricted computational model?

Algebraic Group Model (AGM)

- Algebraic Group Model (AGM)
  recently defined in [Fuchsbauer-Kiltz-Loss@Crypto'18], where an algebraic adversary
  - is able to exploit group specific structures unlike the GGM,
  - is able to compute group operation of given elements like the GGM.

Algebraic Group Model (AGM)

- Algebraic Group Model (AGM)
  recently defined in [Fuchsbauer-Kiltz-Loss@Crypto'18], where an algebraic adversary
  - is able to exploit group specific structures unlike the GGM,
  - is able to compute group operation of given elements like the GGM.

$$\boxed{\begin{array}{l}(\mathbb{G}, g, p) \leftarrow \mathcal{G}(1^\lambda) \text{ and} \\ (X_i = g^{x_i}); (x_1, x_2) \leftarrow \mathbb{Z}_p^2\end{array}} \xrightarrow{(\mathbb{G}, g, p, X_1, X_2)}$$

Algebraic Group Model (AGM)

- Algebraic Group Model (AGM) recently defined in [Fuchsbauer-Kiltz-Loss@Crypto'18], where an algebraic adversary
  - is able to exploit group specific structures unlike the GGM,
  - is able to compute group operation of given elements like the GGM.

$(\mathbb{G}, g, p) \leftarrow \mathcal{G}(1^\lambda)$ and
$(X_i = g^{x_i}); (x_1, x_2) \leftarrow \mathbb{Z}_p^2$

$(\mathbb{G}, g, p, X_1, X_2)$

$Z \leftarrow g^{c_0} X_1^{c_1} X_2^{c_2}$
$\vec{c} = (c_0, c_1, c_2) \in \mathbb{Z}_p^3$

Algebraic Group Model (AGM)

- Algebraic Group Model (AGM)
  recently defined in [Fuchsbauer-Kiltz-Loss@Crypto'18], where an algebraic adversary
  - is able to exploit group specific structures unlike the GGM,
  - is able to compute group operation of given elements like the GGM.

$(\mathbb{G}, g, p) \leftarrow \mathcal{G}(1^\lambda)$ and
$(X_i = g^{x_i}); (x_1, x_2) \leftarrow \mathbb{Z}_p^2$

$(\mathbb{G}, g, p, X_1, X_2)$

$(Z, \vec{c})$

$Z \leftarrow g^{c_0} X_1^{c_1} X_2^{c_2}$
$\vec{c} = (c_0, c_1, c_2) \in \mathbb{Z}_p^3$

Algebraic Group Model (AGM)

- Algebraic Group Model (AGM)
  recently defined in [Fuchsbauer-Kiltz-Loss@Crypto'18], where an algebraic adversary
  - is able to exploit group specific structures unlike the GGM,
  - is able to compute group operation of given elements like the GGM.

$(\mathbb{G}, g, p) \leftarrow \mathcal{G}(1^\lambda)$ and
$(X_i = g^{x_i}); (x_1, x_2) \leftarrow \mathbb{Z}_p^2$

$(\mathbb{G}, g, p, X_1, X_2)$ $\longrightarrow$

$\longleftarrow$ $(Z, \vec{c})$

$Z \leftarrow g^{c_0} X_1^{c_1} X_2^{c_2}$
$\vec{c} = (c_0, c_1, c_2) \in \mathbb{Z}_p^3$

- Advantage of AGM
  - AGM lies in between the standard model and the GGM.
  - AGM is able to derive information theoretic lower bounds in the GGM under *simple* analysis.

Previous Results

- [Fuchsbauer-Kiltz-Loss@Crypto'18]
  - *Tight reduction from the DL to the CDH*
  - (Non-tight) reduction from the DL to the (interactive) strong Diffie-Hellman problem (equivalent to IND-CCA security of Hashed ElGamal encryption scheme in the ROM)
  - (Non-tight) reduction from the DL to the (interactive) LRSW problem (equivalent to UF-CMA security of Camenisch-Lysyanskaya signatures)
  - IND-CCA1 security of the ElGamal encryption scheme under the q-DDH assumption
  - UF-CMA security of Boneh-Lynn-Shacham signatures under the DL assumption
  - The security of Groth's ZK-SNARK under the q-DL assumption

Previous Results

- [Fuchsbauer-Kiltz-Loss@Crypto'18]
  - *Tight reduction from the DL to the CDH*
  - (Non-tight) reduction from the DL to the (interactive) strong Diffie-Hellman problem (equivalent to IND-CCA security of Hashed ElGamal encryption scheme in the ROM)
  - (Non-tight) reduction from the DL to the (interactive) LRSW problem (equivalent to UF-CMA security of Camenisch-Lysyanskaya signatures)
  - IND-CCA1 security of the ElGamal encryption scheme under the q-DDH assumption
  - UF-CMA security of Boneh-Lynn-Shacham signatures under the DL assumption
  - The security of Groth's ZK-SNARK under the q-DL assumption

Can we obtain similar results for other computational problems?

Our Results

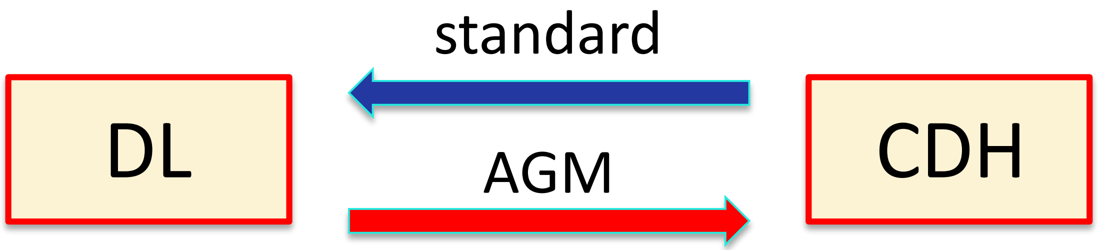
- We successfully provide *tight* reductions from the DL to the CDH, *k*-EDH, *k*-PDH, *k*-linear, BDH.
- We formalize master theorems to indicate when our technique is applicable.
- The tightness is not an advantage but a limitation of our results.
- We continue the research for the Matrix CDH and the kernel Matrix DH.

Our Results

- We successfully provide *tight* reductions from the DL to the CDH, k-EDH, k-PDH, k-linear, BDH.
- We formalize master theorems to indicate when our technique is applicable.
- The tightness is not an advantage but a limitation of our results.
- We continue the research for the Matrix CDH and the kernel Matrix DH.

➢ FKL Reduction

➢ Our Reduction



FKL Reduction: DL to CDH

- Constructing a DL algorithm by using a *SDH* algorithm in the AGM only once

$(\mathbb{G}, g, p) \leftarrow \mathcal{G}(1^\lambda)$ and
$X = g^x; x \leftarrow \mathbb{Z}_p$

$$\xrightarrow{(\mathbb{G}, g, p, X)}$$

FKL Reduction: DL to CDH

- Constructing a DL algorithm by using a *SDH* algorithm in the AGM only once

$(\mathbb{G}, g, p) \leftarrow \mathcal{G}(1^\lambda)$ and
$X = g^x; x \leftarrow \mathbb{Z}_p$

$(\mathbb{G}, g, p, X)$

$(\mathbb{G}, g, p, X)$

FKL Reduction: DL to CDH

- Constructing a DL algorithm by using a *SDH* algorithm in the AGM only once

$(\mathbb{G}, g, p) \leftarrow \mathcal{G}(1^\lambda)$ and
$X = g^x; x \leftarrow \mathbb{Z}_p$

$(\mathbb{G}, g, p, X)$

$(\mathbb{G}, g, p, X)$

$Z \leftarrow g^{c_0} X^{c_1}$
$\vec{c} = (c_0, c_1) \in \mathbb{Z}_p^2$

FKL Reduction: DL to CDH

- Constructing a DL algorithm by using a *SDH* algorithm in the AGM only once

$(\mathbb{G}, g, p) \leftarrow \mathcal{G}(1^\lambda)$ and
$X = g^x; x \leftarrow \mathbb{Z}_p$

$\xrightarrow{(\mathbb{G}, g, p, X)}$

$\xrightarrow{(\mathbb{G}, g, p, X)}$

$\xleftarrow{(Z, \vec{c})}$

$Z \leftarrow g^{c_0} X^{c_1}$
$\vec{c} = (c_0, c_1) \in \mathbb{Z}_p^2$

FKL Reduction: DL to CDH

- Constructing a DL algorithm by using a *SDH* algorithm in the AGM only once

$(\mathbb{G}, g, p) \leftarrow \mathcal{G}(1^\lambda)$ and
$X = g^x; x \leftarrow \mathbb{Z}_p$

$(\mathbb{G}, g, p, X)$

$(\mathbb{G}, g, p, X)$

$(Z, \vec{c})$

$Z \leftarrow g^{c_0} X^{c_1}$
$\vec{c} = (c_0, c_1) \in \mathbb{Z}_p^2$

$$g^{x^2} = g^{c_0} X^{c_1} = g^{c_0 + c_1 x}$$

FKL Reduction: DL to CDH

➢ Constructing a DL algorithm by using a *SDH* algorithm in the AGM only once

$(\mathbb{G}, g, p) \leftarrow \mathcal{G}(1^\lambda)$ and
$X = g^x; x \leftarrow \mathbb{Z}_p$

$\xrightarrow{(\mathbb{G}, g, p, X)}$

$\xrightarrow{(\mathbb{G}, g, p, X)}$

$\xleftarrow{(Z, \vec{c})}$

$Z \leftarrow g^{c_0} X^{c_1}$
$\vec{c} = (c_0, c_1) \in \mathbb{Z}_p^2$

$$g^{x^2} = g^{c_0} X^{c_1} = g^{c_0 + c_1 x}$$

$$x^2 - c_1 x - c_0 = 0 \mod p$$

FKL Reduction: DL to CDH

- Constructing a DL algorithm by using a *SDH* algorithm in the AGM only once

$(\mathbb{G}, g, p) \leftarrow \mathcal{G}(1^\lambda)$ and
$X = g^x; x \leftarrow \mathbb{Z}_p$

$(\mathbb{G}, g, p, X)$ →

← x

$(\mathbb{G}, g, p, X)$ →

← $(Z, \vec{c})$

$Z \leftarrow g^{c_0} X^{c_1}$
$\vec{c} = (c_0, c_1) \in \mathbb{Z}_p^2$

$$g^{x^2} = g^{c_0} X^{c_1} = g^{c_0 + c_1 x}$$

$$x^2 - c_1 x - c_0 = 0 \mod p$$

FKL Reduction: DL to CDH

- Constructing a DL algorithm by using a <u>*SDH*</u> algorithm in the AGM only once

$(\mathbb{G}, g, p) \leftarrow \mathcal{G}(1^\lambda)$ and
$X = g^x; x \leftarrow \mathbb{Z}_p$

$\xrightarrow{(\mathbb{G}, g, p, X)}$

$\xleftarrow{x}$

$\xrightarrow{(\mathbb{G}, g, p, X)}$

$\xleftarrow{(Z, \vec{c})}$

$Z \leftarrow g^{c_0} X^{c_1}$
$\vec{c} = (c_0, c_1) \in \mathbb{Z}_p^2$

$$g^{x^2} = g^{c_0} X^{c_1} = g^{c_0 + c_1 x}$$

$$x^2 - c_1 x - c_0 = 0 \mod p$$

- The reduction in the AGM implies the computational equivalence between the DL and the CDH.
- Information theoretic lower bounds of the CDH in the GGM is $O(\sqrt{p})$ group operations since the above reduction is tight.

Our Direct Reduction: DL to CDH

- Constructing a DL algorithm by using a ***CDH*** algorithm in the AGM only once

$(\mathbb{G}, g, p) \leftarrow \mathcal{G}(1^\lambda)$ and
$X = g^x; x \leftarrow \mathbb{Z}_p$

$(\mathbb{G}, g, p, X)$

Our Direct Reduction: DL to CDH

- Constructing a DL algorithm by using a *CDH* algorithm in the AGM only once

$(\mathbb{G}, g, p) \leftarrow \mathcal{G}(1^\lambda)$ and
$X = g^x; x \leftarrow \mathbb{Z}_p$

$(\mathbb{G}, g, p, X)$

$$a \leftarrow \mathbb{Z}_p$$
$$X_1 \leftarrow X = g^x, X_2 \leftarrow X \cdot g^a = g^{x+a}$$

Our Direct Reduction: DL to CDH

- Constructing a DL algorithm by using a <u>*CDH*</u> algorithm in the AGM only once

$(\mathbb{G}, g, p) \leftarrow \mathcal{G}(1^\lambda)$ and
$X = g^x; x \leftarrow \mathbb{Z}_p$

$\xrightarrow{(\mathbb{G}, g, p, X)}$

$\xrightarrow{(\mathbb{G}, g, p, X_1, X_2)}$

$a \leftarrow \mathbb{Z}_p$
$X_1 \leftarrow X = g^x, X_2 \leftarrow X \cdot g^a = g^{x+a}$

Our Direct Reduction: DL to CDH

- Constructing a DL algorithm by using a *CDH* algorithm in the AGM only once

$(\mathbb{G}, g, p) \leftarrow \mathcal{G}(1^\lambda)$ and
$X = g^x; x \leftarrow \mathbb{Z}_p$

$(\mathbb{G}, g, p, X)$

$(\mathbb{G}, g, p, X_1, X_2)$

$Z \leftarrow g^{c_0} X_1^{c_1} X_2^{c_2}$
$\vec{c} = (c_0, c_1, c_2) \in \mathbb{Z}_p^3$

$a \leftarrow \mathbb{Z}_p$
$X_1 \leftarrow X = g^x, X_2 \leftarrow X \cdot g^a = g^{x+a}$

Our Direct Reduction: DL to CDH

- Constructing a DL algorithm by using a *CDH* algorithm in the AGM only once

$(\mathbb{G}, g, p) \leftarrow \mathcal{G}(1^\lambda)$ and
$X = g^x; x \leftarrow \mathbb{Z}_p$

$\xrightarrow{(\mathbb{G}, g, p, X)}$

$\xrightarrow{(\mathbb{G}, g, p, X_1, X_2)}$

$\xleftarrow{(Z, \vec{c})}$

$Z \leftarrow g^{c_0} X_1^{c_1} X_2^{c_2}$
$\vec{c} = (c_0, c_1, c_2) \in \mathbb{Z}_p^3$

$a \leftarrow \mathbb{Z}_p$
$X_1 \leftarrow X = g^x, X_2 \leftarrow X \cdot g^a = g^{x+a}$

Our Direct Reduction: DL to CDH

- Constructing a DL algorithm by using a *CDH* algorithm in the AGM only once

$(\mathbb{G}, g, p) \leftarrow \mathcal{G}(1^\lambda)$ and
$X = g^x; x \leftarrow \mathbb{Z}_p$

$(\mathbb{G}, g, p, X)$

$(\mathbb{G}, g, p, X_1, X_2)$

$(Z, \vec{c})$

$Z \leftarrow g^{c_0} X_1^{c_1} X_2^{c_2}$
$\vec{c} = (c_0, c_1, c_2) \in \mathbb{Z}_p^3$

$$a \leftarrow \mathbb{Z}_p$$
$$X_1 \leftarrow X = g^x, X_2 \leftarrow X \cdot g^a = g^{x+a}$$

$$g^{x_1 x_2} = g^{c_0} X_1^{c_1} X_2^{c_2}$$
$$= g^{c_0 + c_1 x_1 + c_2 x_2}$$
$$g^{x(x+a)} = g^{c_0 + c_1 x + c_2 (x+a)}$$

Our Direct Reduction: DL to CDH

- Constructing a DL algorithm by using a *CDH* algorithm in the AGM only once

$(\mathbb{G}, g, p) \leftarrow \mathcal{G}(1^\lambda)$ and
$X = g^x; x \leftarrow \mathbb{Z}_p$

$(\mathbb{G}, g, p, X)$

$(\mathbb{G}, g, p, X_1, X_2)$

$(Z, \vec{c})$

$Z \leftarrow g^{c_0} X_1^{c_1} X_2^{c_2}$
$\vec{c} = (c_0, c_1, c_2) \in \mathbb{Z}_p^3$

$a \leftarrow \mathbb{Z}_p$
$X_1 \leftarrow X = g^x, X_2 \leftarrow X \cdot g^a = g^{x+a}$

$$g^{x_1 x_2} = g^{c_0} X_1^{c_1} X_2^{c_2} = g^{c_0 + c_1 x_1 + c_2 x_2}$$
$$g^{x(x+a)} = g^{c_0 + c_1 x + c_2 (x+a)}$$

$$x(x+a) = c_0 + c_1 x + c_2(x+a) \mod p$$
$$\Leftrightarrow x^2 + (a - c_1 - c_2)x - (c_0 + a c_2) = 0 \mod p$$

Our Direct Reduction: DL to CDH

- Constructing a DL algorithm by using a *CDH* algorithm in the AGM only once

$(\mathbb{G}, g, p) \leftarrow \mathcal{G}(1^\lambda)$ and
$X = g^x; x \leftarrow \mathbb{Z}_p$

$(\mathbb{G}, g, p, X)$ →

← x

$(\mathbb{G}, g, p, X_1, X_2)$ →

← $(Z, \vec{c})$

$Z \leftarrow g^{c_0} X_1^{c_1} X_2^{c_2}$
$\vec{c} = (c_0, c_1, c_2) \in \mathbb{Z}_p^3$

$$a \leftarrow \mathbb{Z}_p$$
$$X_1 \leftarrow X = g^x, X_2 \leftarrow X \cdot g^a = g^{x+a}$$

$$g^{x_1 x_2} = g^{c_0} X_1^{c_1} X_2^{c_2}$$
$$= g^{c_0 + c_1 x_1 + c_2 x_2}$$
$$g^{x(x+a)} = g^{c_0 + c_1 x + c_2 (x+a)}$$

$$x(x+a) = c_0 + c_1 x + c_2 (x+a) \mod p$$
$$\Leftrightarrow x^2 + (a - c_1 - c_2)x - (c_0 + a c_2) = 0 \mod p$$

Our Direct Reduction: DL to *k*-EDH

- Constructing a DL algorithm by using a *k-EDH* algorithm in the AGM only once

$(\mathbb{G}, g, p) \leftarrow \mathcal{G}(1^\lambda)$ and
$X = g^x; x \leftarrow \mathbb{Z}_p$

$\xrightarrow{(\mathbb{G}, g, p, X)}$

$\xleftarrow{x}$

$\xrightarrow{(\mathbb{G}, g, p, X)}$

$\xleftarrow{(Z, \vec{c})}$

$Z \leftarrow g^{c_0} X^{c_1}$
$\vec{c} = (c_0, c_1) \in \mathbb{Z}_p^2$

$$g^{x^k} = g^{c_0} X^{c_1} = g^{c_0 + c_1 x}$$

$$x^k = c_1 x + c_0 \mod p$$

Our Direct Reduction: DL to *k*-PDH

- Constructing a DL algorithm by using a *k-PDH* algorithm in the AGM only once

$(\mathbb{G}, g, p) \leftarrow \mathcal{G}(1^\lambda)$ and

$X = g^x; x \leftarrow \mathbb{Z}_p$

$$(\mathbb{G}, g, p, X) \longrightarrow$$

$$\longleftarrow x$$

$$(\mathbb{G}, g, p, (X_i)\ ) \longrightarrow$$

$$\longleftarrow (Z, \vec{c})$$

$Z \leftarrow g^{c_0} X_1^{c_1} \cdots X_k^{c_k}$

$\vec{c} = (c_0, c_1, \ldots, c_k)$

$\in \mathbb{Z}_p^{k+1}$

$$(a_2, \ldots, a_k) \leftarrow \mathbb{Z}_p^{k-1}$$

$$X_1 \leftarrow X, X_i \leftarrow X \cdot g^{a_i} = g^{x+a_i}$$

$$g^{x_1 \cdots x_k} = g^{c_0} X_1^{c_1} \cdots X_k^{c_k}$$
$$= g^{c_0 + c_1 x_1 + \cdots + c_k x_k}$$

$$g^{x(x+a_2)\cdots(x+a_k)}$$
$$= g^{c_0 + c_1 x + c_2(x+a_2) + \cdots + c_k(x+a_k)}$$

$$x(x+a_2)\cdots(x+a_k) = c_0 + c_1 x + c_2(x+a) + \cdots + c_k(x+a_k) \mod p$$

Our Direct Reduction: DL to BDH

- Constructing a DL algorithm by using a <u>*BDH*</u> algorithm in the AGM only once

$$(\mathbb{G}, \mathbb{G}_T, g, e, p) \leftarrow \mathcal{G}(1^\lambda)$$
$$X = g^x; x \leftarrow \mathbb{Z}_p$$

$$\xrightarrow{(\mathbb{G}, \mathbb{G}_T, g, e, p, X)}$$

Our Direct Reduction: DL to BDH

- Constructing a DL algorithm by using a *BDH* algorithm in the AGM only once

$(\mathbb{G}, \mathbb{G}_T, g, e, p) \leftarrow \mathcal{G}(1^\lambda)$
$X = g^x; x \leftarrow \mathbb{Z}_p$

$\xrightarrow{(\mathbb{G}, \mathbb{G}_T, g, e, p, X)}$

$\xrightarrow{(\mathbb{G}, \mathbb{G}_T, g, e, p, (X_i))}$

$(a_2, a_3) \leftarrow \mathbb{Z}_p^2$
$X_1 \leftarrow X, X_i \leftarrow X \cdot g^{a_i} = g^{x+a_i}$

Our Direct Reduction: DL to BDH

- Constructing a DL algorithm by using a *BDH* algorithm in the AGM only once

$(\mathbb{G}, \mathbb{G}_T, g, e, p) \leftarrow \mathcal{G}(1^\lambda)$
$X = g^x; x \leftarrow \mathbb{Z}_p$

$(\mathbb{G}, \mathbb{G}_T, g, e, p, X)$ →

$(\mathbb{G}, \mathbb{G}_T, g, e, p, (X_i))$ →

← $(Z, \vec{c})$

$Z \leftarrow e(g,g)^{c_0} e(g, X_1)^{c_1}$
$\cdots$
$\vec{c} = (c_0, c_1, \ldots) \in \mathbb{Z}_p^{10}$

$(a_2, a_3) \leftarrow \mathbb{Z}_p^2$
$X_1 \leftarrow X, X_i \leftarrow X \cdot g^{a_i} = g^{x+a_i}$

Our Direct Reduction: DL to BDH

- Constructing a DL algorithm by using a <u>*BDH*</u> algorithm in the AGM only once

$(\mathbb{G}, \mathbb{G}_T, g, e, p, X)$

$(\mathbb{G}, \mathbb{G}_T, g, e, p) \leftarrow \mathcal{G}(1^\lambda)$
$X = g^x; x \leftarrow \mathbb{Z}_p$

$(\mathbb{G}, \mathbb{G}_T, g, e, p, (X_i))$

$(Z, \vec{c})$

$Z \leftarrow e(g,g)^{c_0} e(g, X_1)^{c_1}$
$\cdots$
$\vec{c} = (c_0, c_1, \dots) \in \mathbb{Z}_p^{10}$

$$(a_2, a_3) \leftarrow \mathbb{Z}_p^2$$
$$X_1 \leftarrow X, X_i \leftarrow X \cdot g^{a_i} = g^{x+a_i}$$

$$e(g,g)^{xyz} = e(g,g)^{c_0} e(g,X_1)^{c_1} e(g,X_2)^{c_2} e(g,X_3)^{c_3} e(X_1,X_1)^{c_4} e(X_1,X_2)^{c_5} e(X_1,X_3)^{c_6}$$
$$e(X_2,X_2)^{c_7} e(X_2,X_3)^{c_8} e(X_3,X_3)^{c_9}$$

$$x(x+a_2)(x+a_3) = c_0 + c_1 x + c_2(x+a_2) + c_3(x+a_3) + c_4 x^2 + c_5 x(x+a_2) + c_6 x(x+a_3)$$
$$+c_7(x+a_2)^2 + c_8(x+a_2)(x+a_3) + c_9(x+a_3)^2 \mod p$$

Our Direct Reduction: DL to BDH

- Constructing a DL algorithm by using a *BDH* algorithm in the AGM only once

$$(\mathbb{G}, \mathbb{G}_T, g, e, p) \leftarrow \mathcal{G}(1^\lambda)$$
$$X = g^x; x \leftarrow \mathbb{Z}_p$$

$$\xrightarrow{(\mathbb{G}, \mathbb{G}_T, g, e, p, X)}$$
$$\xleftarrow{x}$$

$$\xrightarrow{(\mathbb{G}, \mathbb{G}_T, g, e, p, (X_i))}$$
$$\xleftarrow{(Z, \vec{c})}$$

$$Z \leftarrow e(g,g)^{c_0} e(g, X_1)^{c_1}$$
$$\cdots$$
$$\vec{c} = (c_0, c_1, \dots) \in \mathbb{Z}_p^{10}$$

$$(a_2, a_3) \leftarrow \mathbb{Z}_p^2$$
$$X_1 \leftarrow X, X_i \leftarrow X \cdot g^{a_i} = g^{x+a_i}$$

$$e(g,g)^{xyz} = e(g,g)^{c_0} e(g,X_1)^{c_1} e(g,X_2)^{c_2} e(g,X_3)^{c_3} e(X_1,X_1)^{c_4} e(X_1,X_2)^{c_5} e(X_1,X_3)^{c_6}$$
$$e(X_2,X_2)^{c_7} e(X_2,X_3)^{c_8} e(X_3,X_3)^{c_9}$$

$$x(x+a_2)(x+a_3) = c_0 + c_1 x + c_2(x+a_2) + c_3(x+a_3) + c_4 x^2 + c_5 x(x+a_2) + c_6 x(x+a_3)$$
$$+ c_7(x+a_2)^2 + c_8(x+a_2)(x+a_3) + c_9(x+a_3)^2 \mod p$$

Obtaining Non-zero Modular Equations

$$X_1 = g^x, X_2 = g^{x+a_2}, X_3 = g^{x+a_3}$$

$$\begin{aligned} e(g,g)^{xyz} = {} & e(g,g)^{c_0} e(g,X_1)^{c_1} e(g,X_2)^{c_2} e(g,X_3)^{c_3} e(X_1,X_1)^{c_4} e(X_1,X_2)^{c_5} e(X_1,X_3)^{c_6} \\ & e(X_2,X_2)^{c_7} e(X_2,X_3)^{c_8} e(X_3,X_3)^{c_9} \end{aligned}$$

$$\begin{aligned} x(x+a_2)(x+a_3) = {} & c_0 + c_1 x + c_2(x+a_2) + c_3(x+a_3) + c_4 x^2 + c_5 x(x+a_2) + c_6 x(x+a_3) \\ & + c_7(x+a_2)^2 + c_8(x+a_2)(x+a_3) + c_9(x+a_3)^2 \mod p \end{aligned}$$

Obtaining Non-zero Modular Equations

$$X_1 = g^x, X_2 = g^{x+a_2}, X_3 = g^{x+a_3}$$

$$\begin{aligned} e(g,g)^{xyz} = {} & e(g,g)^{c_0} e(g,X_1)^{c_1} e(g,X_2)^{c_2} e(g,X_3)^{c_3} e(X_1,X_1)^{c_4} e(X_1,X_2)^{c_5} e(X_1,X_3)^{c_6} \\ & e(X_2,X_2)^{c_7} e(X_2,X_3)^{c_8} e(X_3,X_3)^{c_9} \end{aligned}$$

$$\begin{aligned} x(x+a_2)(x+a_3) = {} & c_0 + c_1 x + c_2(x+a_2) + c_3(x+a_3) + c_4 x^2 + c_5 x(x+a_2) + c_6 x(x+a_3) \\ & + c_7(x+a_2)^2 + c_8(x+a_2)(x+a_3) + c_9(x+a_3)^2 \mod p \end{aligned}$$

- The degree of the left hand side of the modular equation is 3.
- The degree of the right hand side of the modular equation is at most 2.

Obtaining Non-zero Modular Equations

$$X_1 = g^x, X_2 = g^{x+a_2}, X_3 = g^{x+a_3}$$

$$\begin{aligned} e(g,g)^{xyz} = {} & e(g,g)^{c_0} e(g,X_1)^{c_1} e(g,X_2)^{c_2} e(g,X_3)^{c_3} e(X_1,X_1)^{c_4} e(X_1,X_2)^{c_5} e(X_1,X_3)^{c_6} \\ & e(X_2,X_2)^{c_7} e(X_2,X_3)^{c_8} e(X_3,X_3)^{c_9} \end{aligned}$$

$$\begin{aligned} x(x+a_2)(x+a_3) = {} & c_0 + c_1 x + c_2(x+a_2) + c_3(x+a_3) + c_4 x^2 + c_5 x(x+a_2) + c_6 x(x+a_3) \\ & + c_7(x+a_2)^2 + c_8(x+a_2)(x+a_3) + c_9(x+a_3)^2 \mod p \end{aligned}$$

- The degree of the left hand side of the modular equation is 3.
- The degree of the right hand side of the modular equation is at most 2.

$$\begin{aligned} & x^3 + (a_2 + a_3 - c_4 - c_5 - c_6 - c_7 - c_8 - c_9)x^2 \\ & +(a_2 a_3 - c_1 - c_2 - c_3 - a_2 c_5 - a_3 c_6 - 2a_2 c_7 - (a_2 + a_3)c_8 - 2a_3 c_9)x \\ & -(c_0 + a_2 c_2 + a_3 c_3 + a_2^2 c_7 + a_2 a_3 c_8 + a_3^2 c_9) = 0 \mod p \end{aligned}$$

Obtaining Non-zero Modular Equations

$$X_1 = g^x, X_2 = g^{x+a_2}, X_3 = g^{x+a_3}$$

$$\begin{aligned} e(g,g)^{xyz} = {} & e(g,g)^{c_0} e(g,X_1)^{c_1} e(g,X_2)^{c_2} e(g,X_3)^{c_3} e(X_1,X_1)^{c_4} e(X_1,X_2)^{c_5} e(X_1,X_3)^{c_6} \\ & e(X_2,X_2)^{c_7} e(X_2,X_3)^{c_8} e(X_3,X_3)^{c_9} \end{aligned}$$

$$\begin{aligned} x(x+a_2)(x+a_3) = {} & c_0 + c_1 x + c_2(x+a_2) + c_3(x+a_3) + c_4 x^2 + c_5 x(x+a_2) + c_6 x(x+a_3) \\ & + c_7(x+a_2)^2 + c_8(x+a_2)(x+a_3) + c_9(x+a_3)^2 \mod p \end{aligned}$$

- The degree of the left hand side of the modular equation is 3.
- The degree of the right hand side of the modular equation is at most 2.

$$\begin{aligned} & x^3 + (a_2 + a_3 - c_4 - c_5 - c_6 - c_7 - c_8 - c_9)x^2 \\ & + (a_2 a_3 - c_1 - c_2 - c_3 - a_2 c_5 - a_3 c_6 - 2a_2 c_7 - (a_2 + a_3)c_8 - 2a_3 c_9)x \\ & - (c_0 + a_2 c_2 + a_3 c_3 + a_2^2 c_7 + a_2 a_3 c_8 + a_3^2 c_9) = 0 \mod p \end{aligned}$$

- The modular polynomial has to be non-zero.

Our Direct Reduction: DL to CDH

- Constructing a DL algorithm by using a *CDH* algorithm in the AGM only once

$(\mathbb{G}, g, p) \leftarrow \mathcal{G}(1^\lambda)$ and
$X = g^x; x \leftarrow \mathbb{Z}_p$

$(\mathbb{G}, g, p, X) \rightarrow$

$\leftarrow x$

$(\mathbb{G}, g, p, X_1, X_2) \rightarrow$

$\leftarrow (Z, \vec{c})$

$Z \leftarrow g^{c_0} X_1^{c_1} X_2^{c_2}$
$\vec{c} = (c_0, c_1, c_2) \in \mathbb{Z}_p^3$

$$a \leftarrow \mathbb{Z}_p$$
$$X_1 \leftarrow X = g^x, X_2 \leftarrow X \cdot g^a = g^{x+a}$$

$$g^{x_1 x_2} = g^{c_0} X_1^{c_1} X_2^{c_2} = g^{c_0 + c_1 x_1 + c_2 x_2}$$
$$g^{x(x+a)} = g^{c_0 + c_1 x + c_2 (x+a)}$$

$$x(x+a) = c_0 + c_1 x + c_2 (x+a) \mod p$$
$$\Leftrightarrow x^2 + (a - c_1 - c_2)x - (c_0 + a c_2) = 0 \mod p$$

Our Direct Reduction: DL to *k*-EDH

- Constructing a DL algorithm by using a *k-EDH* algorithm in the AGM only once

$(\mathbb{G}, g, p) \leftarrow \mathcal{G}(1^\lambda)$ and
$X = g^x; x \leftarrow \mathbb{Z}_p$

$\xrightarrow{(\mathbb{G}, g, p, X)}$

$\xleftarrow{x}$

$\xrightarrow{(\mathbb{G}, g, p, X)}$

$\xleftarrow{(Z, \vec{c})}$

$Z \leftarrow g^{c_0} X^{c_1}$
$\vec{c} = (c_0, c_1) \in \mathbb{Z}_p^2$

$$g^{x^k} = g^{c_0} X^{c_1} = g^{c_0 + c_1 x}$$

$$x^k = c_1 x + c_0 \mod p$$

Our Direct Reduction: DL to *k*-PDH

➢ Constructing a DL algorithm by using a *k-PDH* algorithm in the AGM only once

$(\mathbb{G}, g, p) \leftarrow \mathcal{G}(1^\lambda)$ and
$X = g^x; x \leftarrow \mathbb{Z}_p$

$(\mathbb{G}, g, p, X)$ →

← x

$(\mathbb{G}, g, p, (X_i)\ )$ →

← $(Z, \vec{c})$

$Z \leftarrow g^{c_0} X_1^{c_1} \cdots X_k^{c_k}$
$\vec{c} = (c_0, c_1, \dots, c_k) \in \mathbb{Z}_p^{k+1}$

$$(a_2, \dots, a_k) \leftarrow \mathbb{Z}_p^{k-1}$$
$$X_1 \leftarrow X, X_i \leftarrow X \cdot g^{a_i} = g^{x+a_i}$$

$$g^{x_1 \cdots x_k} = g^{c_0} X_1^{c_1} \cdots X_k^{c_k} = g^{c_0 + c_1 x_1 + \cdots + c_k x_k}$$

$$g^{x(x+a_2)\cdots(x+a_k)} = g^{c_0 + c_1 x + c_2(x+a_2) + \cdots + c_k(x+a_k)}$$

$$x(x + a_2) \cdots (x + a_k) = c_0 + c_1 x + c_2(x + a) + \cdots + c_k(x + a_k) \mod p$$

Master Theorem in Cyclic Groups

- Generalized Computational Diffie-Hellman (GDH) Problem
  Input: $(\mathbb{G}, g, p)$ and $\left(X_i = g^{f_i(x_1,\ldots,x_m,y_1,\ldots,y_n)}\right)$; $(x_1, \ldots, x_m, y_1, \ldots, y_n) \leftarrow \mathbb{Z}_p^{m+n}$
  Solution: $g^{g(x_1,\ldots,x_m)}$

Master Theorem in Cyclic Groups

- Generalized Computational Diffie-Hellman (GDH) Problem
  Input: $(\mathbb{G}, g, p)$ and $\left(X_i = g^{f_i(x_1,\ldots,x_m,y_1,\ldots,y_n)}\right); (x_1, \ldots, x_m, y_1, \ldots, y_n) \leftarrow \mathbb{Z}_p^{m+n}$
  Solution: $g^{g(x_1,\ldots,x_m)}$
- Computational Diffie-Hellman (CDH) Problem
  Input: $(\mathbb{G}, g, p)$ and $(X_1 = g^{x_1}, X_2 = g^{x_2}); (x_1, x_2) \leftarrow \mathbb{Z}_p^2$
  Solution: $g^{x_1 x_2}$
- *k*-Exponent Diffie-Hellman (*k*-EDH) Problem
  Input: $(\mathbb{G}, g, p)$ and $X = g^x; x \leftarrow \mathbb{Z}_p$
  Solution: g^{x^k}
- *k*-Party Diffie-Hellman (*k*-PDH) Problem
  Input: $(\mathbb{G}, g, p)$ and $(X_1 = g^{x_1}, \ldots, X_k = g^{x_k}); (x_1, \ldots, x_k) \leftarrow \mathbb{Z}_p^k$
  Solution: $g^{x_1 \cdots x_k}$

Master Theorem in Cyclic Groups

- Generalized Computational Diffie-Hellman (GDH) Problem
  Input: $(\mathbb{G}, g, p)$ and $\left(X_i = g^{f_i(x_1,\ldots,x_m,y_1,\ldots,y_n)}\right)$; $(x_1, \ldots, x_m, y_1, \ldots, y_n) \leftarrow \mathbb{Z}_p^{m+n}$
  Solution: $g^{g(x_1,\ldots,x_m)}$

Master Theorem in Cyclic Groups

- Generalized Computational Diffie-Hellman (GDH) Problem
  Input: $(\mathbb{G}, g, p)$ and $\left(X_i = g^{f_i(x_1,\ldots,x_m,y_1,\ldots,y_n)}\right)$; $(x_1, \ldots, x_m, y_1, \ldots, y_n) \leftarrow \mathbb{Z}_p^{m+n}$
  Solution: $g^{g(x_1,\ldots,x_m)}$

We can provide a reduction from the DL to the GDH when

- $\deg_{x_1,\ldots,x_m} f_i(x_1, \ldots, x_m, y_1, \ldots, y_n) = 0 \text{ or } 1$
  (to embed the DL solution into GDH instance)
- $\deg g(x_1, \ldots, x_m) \neq 0 \text{ and } 1$
  (so that the modular polynomial is non-zero)

Our Direct Reduction: DL to GDH

- Constructing a DL algorithm by using a ***GDH*** algorithm in the AGM only once

$(\mathbb{G}, g, p) \leftarrow \mathcal{G}(1^\lambda)$ and
$X = g^x; x \leftarrow \mathbb{Z}_p$

$\xrightarrow{(\mathbb{G}, g, p, X)}$

Our Direct Reduction: DL to GDH

- Constructing a DL algorithm by using a *GDH* algorithm in the AGM only once

$(\mathbb{G}, g, p) \leftarrow \mathcal{G}(1^\lambda)$ and
$X = g^x; x \leftarrow \mathbb{Z}_p$

$(\mathbb{G}, g, p, X)$

$(\mathbb{G}, g, p, X_i)$

$$(y_1, \ldots, y_n) \leftarrow \mathbb{Z}_p^n$$
$$(a_1, \ldots, a_{m-1}) \leftarrow \mathbb{Z}_p^m$$
$$x_1 \leftarrow x, x_i \leftarrow x + a_i$$
$$X_i = g^{f_i(x_1, \ldots, x_m, y_1, \ldots, y_n)}$$

Our Direct Reduction: DL to GDH

- Constructing a DL algorithm by using a *GDH* algorithm in the AGM only once

$(\mathbb{G}, g, p) \leftarrow \mathcal{G}(1^\lambda)$ and
$X = g^x; x \leftarrow \mathbb{Z}_p$

$\xrightarrow{(\mathbb{G}, g, p, X)}$

$\xrightarrow{(\mathbb{G}, g, p, X_i)}$

$$(y_1, \dots, y_n) \leftarrow \mathbb{Z}_p^n$$
$$(a_1, \dots, a_{m-1}) \leftarrow \mathbb{Z}_p^m$$
$$x_1 \leftarrow x, x_i \leftarrow x + a_i$$
$$X_i = g^{f_i(x_1, \dots, x_m, y_1, \dots, y_n)}$$

$$\deg_{x_1, \dots, x_m} f_i(x_1, \dots, x_m, y_1, \dots, y_n) \ = 0 \text{ or } 1$$
(to embed the DL solution into GDH instance)

Our Direct Reduction: DL to GDH

- Constructing a DL algorithm by using a *GDH* algorithm in the AGM only once

$(\mathbb{G}, g, p) \leftarrow \mathcal{G}(1^\lambda)$ and
$X = g^x; x \leftarrow \mathbb{Z}_p$

$\xrightarrow{(\mathbb{G}, g, p, X)}$

$\xrightarrow{(\mathbb{G}, g, p, X_i)}$

Our Direct Reduction: DL to GDH

- Constructing a DL algorithm by using a *GDH* algorithm in the AGM only once

$(\mathbb{G}, g, p) \leftarrow \mathcal{G}(1^\lambda)$ and
$X = g^x; x \leftarrow \mathbb{Z}_p$

$(\mathbb{G}, g, p, X)$

$(\mathbb{G}, g, p, X_i)$

$(Z, \vec{c})$

$Z \leftarrow g^{c_0} X_1^{c_1} \cdots$
$\vec{c} = (c_0, c_1, \dots) \in \mathbb{Z}_p^\ell$

$$g(x_1, \dots, x_m) = c_0 + c_i f_i(x_1, \dots, x_m, y_1, \dots, y_n) \mod p$$

Our Direct Reduction: DL to GDH

➢ Constructing a DL algorithm by using a <u>*GDH*</u> algorithm in the AGM only once

$(\mathbb{G}, g, p) \leftarrow \mathcal{G}(1^\lambda)$ and
$X = g^x; x \leftarrow \mathbb{Z}_p$

$\xrightarrow{(\mathbb{G}, g, p, X)}$

$\xrightarrow{(\mathbb{G}, g, p, X_i)}$

$\xleftarrow{(Z, \vec{c})}$

$Z \leftarrow g^{c_0} X_1^{c_1} \cdots$
$\vec{c} = (c_0, c_1, \ldots) \in \mathbb{Z}_p^\ell$

$$g(x_1, \ldots, x_m) = c_0 + c_i f_i(x_1, \ldots, x_m, y_1, \ldots, y_n) \mod p$$

- $\deg_{x_1,\ldots,x_m} f_i(x_1, \ldots, x_m, y_1, \ldots, y_n) = 0 \text{ or } 1$
  (to embed the DL solution into GDH instance)
- $\deg g(x_1, \ldots, x_m) \neq 0 \text{ and } 1$
  (so that the modular polynomial is non-zero)

$\Rightarrow$

The modular polynomial is non-zero!

Our Direct Reduction: DL to GDH

➢ Constructing a DL algorithm by using a *GDH* algorithm in the AGM only once

$(\mathbb{G}, g, p) \leftarrow \mathcal{G}(1^\lambda)$ and
$X = g^x; x \leftarrow \mathbb{Z}_p$

$(\mathbb{G}, g, p, X) \rightarrow$

$\leftarrow x$

$(\mathbb{G}, g, p, X_i) \rightarrow$

$\leftarrow (Z, \vec{c})$

$Z \leftarrow g^{c_0} X_1^{c_1} \cdots$
$\vec{c} = (c_0, c_1, \dots) \in \mathbb{Z}_p^\ell$

$$g(x_1, \dots, x_m) = c_0 + c_i f_i(x_1, \dots, x_m, y_1, \dots, y_n) \mod p$$

- $\deg_{x_1,\dots,x_m} f_i(x_1, \dots, x_m, y_1, \dots, y_n) = 0$ or 1
  (to embed the DL solution into GDH instance)
- $\deg g(x_1, \dots, x_m) \neq 0$ and 1
  (so that the modular polynomial is non-zero)

⇒ The modular polynomial is non-zero!

Conclusion

- We provided tight reductions from the DL to several variants of Diffie-Hellman problems in the AGM defined by [FKL@Crypto'18].
- We define the AGM in symmetric bilinear groups by following [FKL@Crypto'18]'s definition in cyclic groups.
- We formalized *master theorems* to indicate the Diffie-Hellman variants that can be reduced to from the DL by following our approach.
- Our master theorem does not include the *k*-linear problem. Therefore, we provided an tailor-made reduction for the *k*-linear problem.
- As future works, we try to study the Matrix CDH and the Kernel Matrix DH.
- Other interesting future works are analogous results in *composite*-order groups, *decision* problems, or non-tight reductions.