A Classification of Computational Assumptions in the Algebraic Group Model

Balthazar Bauer, Georg Fuchsbauer, Julian Loss

August 11, 2020

1. The Algebraic Group Model (FKL 2018)

2. Classification

3. Separation

1. The Algebraic Group Model (FKL 2018)
2. Classification
3. Separation

From GGM to AGM

- Let $\mathbb{G}$ be a cyclic group of prime order p.

From GGM to AGM

- Let $\mathbb{G}$ be a cyclic group of prime order p.
  Standard Model

$$\mathcal{C} \xrightarrow{(\mathbf{Z}_1, \mathbf{Z}_2, \mathbf{Z}_3)}$$

From GGM to AGM

- Let $\mathbb{G}$ be a cyclic group of prime order p.
  Standard Model

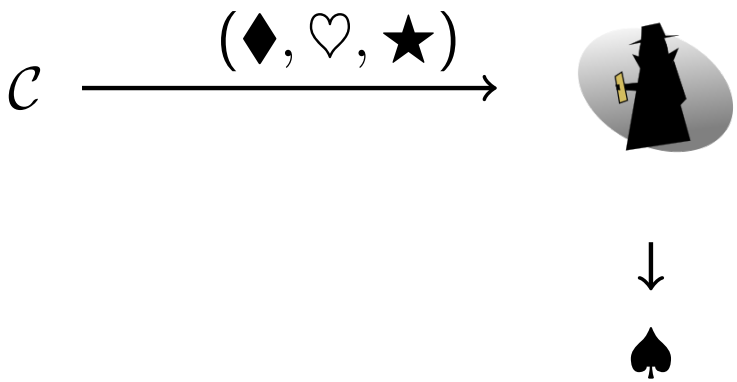
$$\mathcal{C} \xrightarrow{(\mathbf{Z}_1, \mathbf{Z}_2, \mathbf{Z}_3)}$$

$$\downarrow$$

$$\mathbf{Y}$$

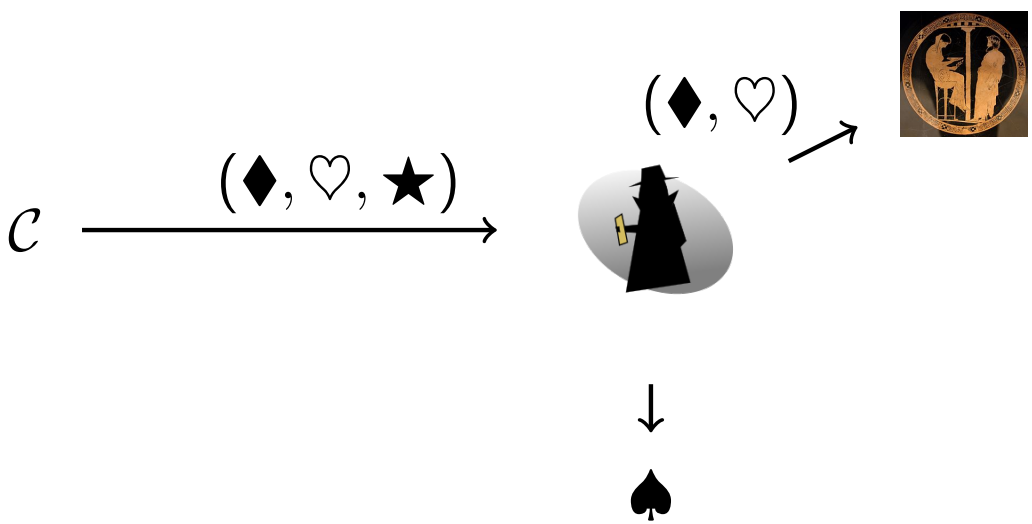
From GGM to AGM

- Let $\mathbb{G}$ be a cyclic group of prime order p.
  Generic Group Model



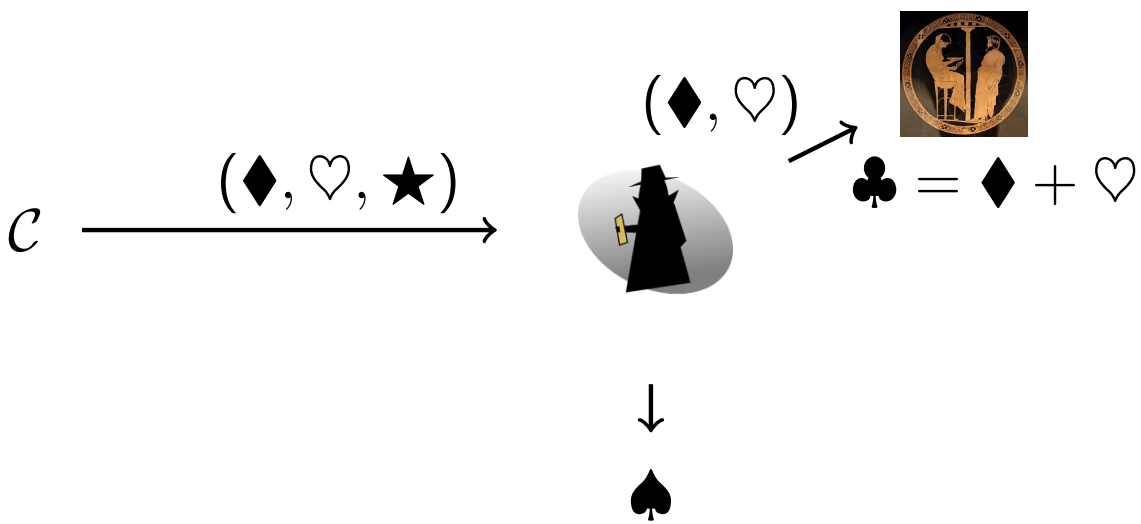
From GGM to AGM

- Let $\mathbb{G}$ be a cyclic group of prime order p.
  Generic Group Model



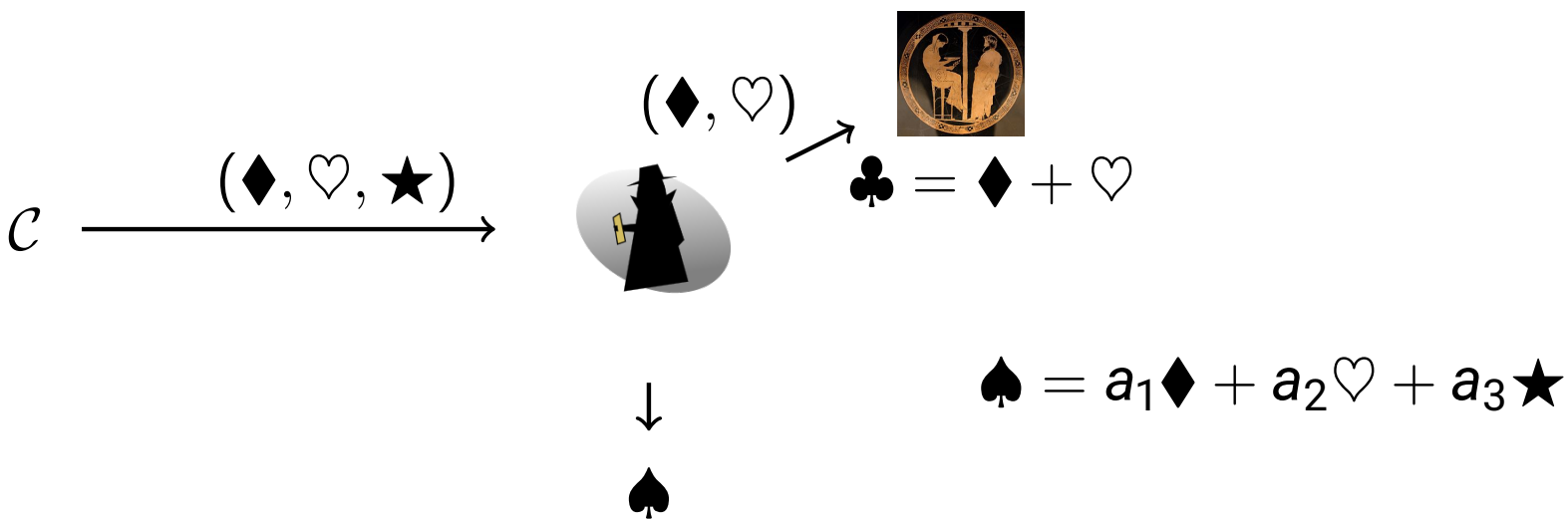
From GGM to AGM

- Let $\mathbb{G}$ be a cyclic group of prime order p.
  Generic Group Model



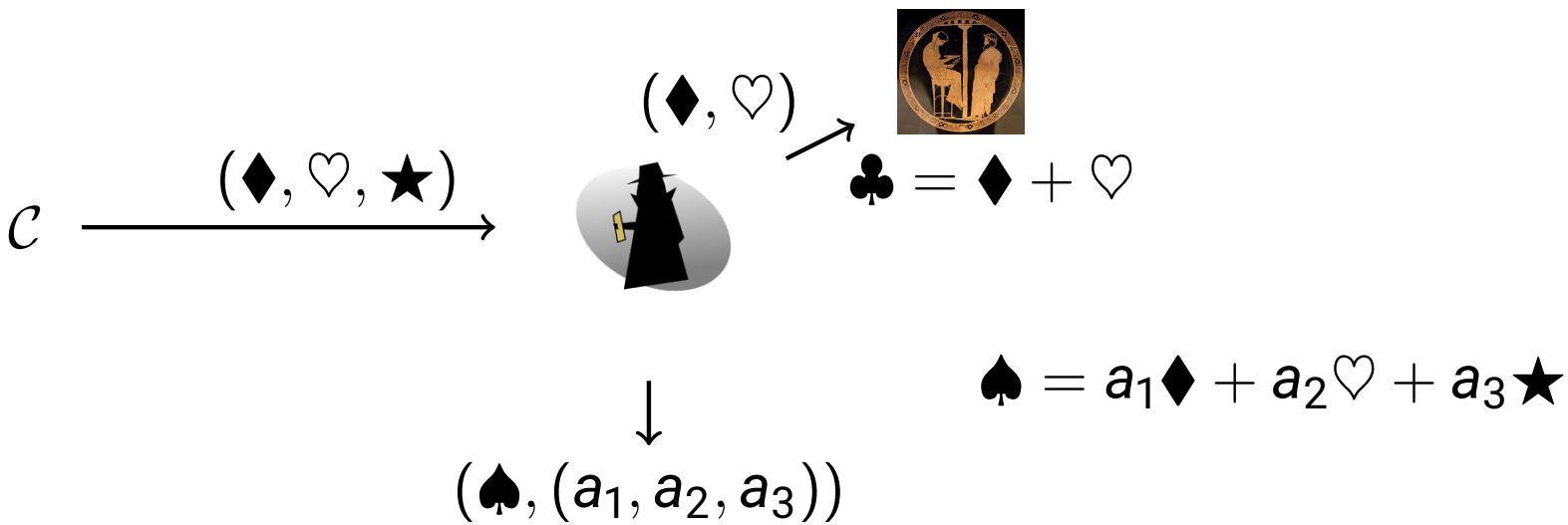
From GGM to AGM

- Let $\mathbb{G}$ be a cyclic group of prime order p.
  Generic Group Model



From GGM to AGM

- Let $\mathbb{G}$ be a cyclic group of prime order p.
  Generic Group Model (modified)



From GGM to AGM

- Let $\mathbb{G}$ be a cyclic group of prime order p.
  Algebraic Group Model

$$\mathcal{C} \xrightarrow{(\mathbf{Z}_1, \mathbf{Z}_2, \mathbf{Z}_3)}$$

$$\downarrow$$

$$(\mathbf{Y}, (a_1, a_2, a_3))$$

$$\mathbf{Y} = a_1\mathbf{Z}_1 + a_2\mathbf{Z}_2 + a_3\mathbf{Z}_3$$

Standard vs Algebraic

- No reduction from DLog to CDH in the standard model.

Standard vs Algebraic

- No reduction from DLog to CDH in the standard model.
- Let $\mathcal{A}$ be an algebraic algorithm which solves CDH.

Standard vs Algebraic

- No reduction from DLog to CDH in the standard model.
- Let $\mathcal{A}$ be an algebraic algorithm which solves CDH.

- $\mathcal{B}(\mathbf{G}, \mathbf{X})$:

Standard vs Algebraic

- No reduction from DLog to CDH in the standard model.
- Let $\mathcal{A}$ be an algebraic algorithm which solves CDH.

- $\mathcal{B}(\mathbf{G}, \mathbf{X})$:
  - $v \xleftarrow{\$} \mathbb{Z}_p^*$

Standard vs Algebraic

- No reduction from DLog to CDH in the standard model.
- Let $\mathcal{A}$ be an algebraic algorithm which solves CDH.

- $\mathcal{B}(\mathbf{G}, \mathbf{X})$:
  - $v \xleftarrow{\$} \mathbb{Z}_p^*$
  - $(\mathbf{Y}, \ell_1, \ell_2, \ell_3) \leftarrow \mathcal{A}(\mathbf{G}, \mathbf{X}, \mathbf{X} + v\mathbf{G})$

Standard vs Algebraic

- No reduction from DLog to CDH in the standard model.
- Let $\mathcal{A}$ be an algebraic algorithm which solves CDH.

- $\mathcal{B}(\mathbf{G}, \mathbf{X})$:
  - $v \xleftarrow{\$} \mathbb{Z}_p^*$
  - $(\mathbf{Y}, \ell_1, \ell_2, \ell_3) \leftarrow \mathcal{A}(\mathbf{G}, \mathbf{X}, \mathbf{X} + v\mathbf{G})$
    $(\ell_1\mathbf{G} + \ell_2\mathbf{X} + \ell_3(\mathbf{X} + v\mathbf{G}) = \mathbf{Y})$

Standard vs Algebraic

- No reduction from DLog to CDH in the standard model.
- Let $\mathcal{A}$ be an algebraic algorithm which solves CDH.

- $\mathcal{B}(\mathbf{G}, \mathbf{X})$:
  - $v \xleftarrow{\$} \mathbb{Z}_p^*$
  - $(\mathbf{Y}, \ell_1, \ell_2, \ell_3) \leftarrow \mathcal{A}(\mathbf{G}, \mathbf{X}, \mathbf{X} + v\mathbf{G})$
    $(\ell_1\mathbf{G} + \ell_2\mathbf{X} + \ell_3(\mathbf{X} + v\mathbf{G}) = \mathbf{Y})$
  - $\{x_1^*, x_2^*\} \leftarrow$ Solve $(\ell_1 + \ell_2 X + \ell_3(X + v)) \equiv X(X + v) \pmod{p}$

Standard vs Algebraic

- No reduction from DLog to CDH in the standard model.
- Let $\mathcal{A}$ be an algebraic algorithm which solves CDH.

- $\mathcal{B}(\mathbf{G}, \mathbf{X})$:
  - $v \xleftarrow{\$} \mathbb{Z}_p^*$
  - $(\mathbf{Y}, \ell_1, \ell_2, \ell_3) \leftarrow \mathcal{A}(\mathbf{G}, \mathbf{X}, \mathbf{X} + v\mathbf{G})$
    $(\ell_1\mathbf{G} + \ell_2\mathbf{X} + \ell_3(\mathbf{X} + v\mathbf{G}) = \mathbf{Y})$
  - $\{x_1^*, x_2^*\} \leftarrow$ Solve $(\ell_1 + \ell_2 X + \ell_3(X + v)) \equiv X(X + v) \pmod{p}$
  - Output x_i^* such that $\mathbf{X} = x_i^*\mathbf{G}$

Standard vs Algebraic

- No reduction from DLog to CDH in the standard model.
- Let $\mathcal{A}$ be an algebraic algorithm which solves CDH.

- $\mathcal{B}(\mathbf{G}, \mathbf{X})$:
  - $v \xleftarrow{\$} \mathbb{Z}_p^*$
  - $(\mathbf{Y}, \ell_1, \ell_2, \ell_3) \leftarrow \mathcal{A}(\mathbf{G}, \mathbf{X}, \mathbf{X} + v\mathbf{G})$
    $(\ell_1\mathbf{G} + \ell_2\mathbf{X} + \ell_3(\mathbf{X} + v\mathbf{G}) = \mathbf{Y})$
  - $\{x_1^*, x_2^*\} \leftarrow$ Solve $(\ell_1 + \ell_2 X + \ell_3(X + v)) \equiv X(X + v) \pmod{p}$
  - Output x_i^* such that $\mathbf{X} = x_i^*\mathbf{G}$
- Conclusion: AGM enables new security reductions

q-Diffie-Hellman Exponent

- Let $\mathbb{G}$ be a cyclic group of prime order p.

q-Diffie-Hellman Exponent

- Let $\mathbb{G}$ be a cyclic group of prime order p.

$$x \xleftarrow{\$} \mathbb{Z}_p; \begin{pmatrix} \mathbf{G} \\ x\mathbf{G} \\ x^2\mathbf{G} \\ . \\ . \\ x^q\mathbf{G} \end{pmatrix} \rightarrow \quad \rightarrow x^{q+1}\mathbf{G}$$

q-Diffie-Hellman Exponent

- Let $\mathbb{G}$ be a cyclic group of prime order p.

$$x \xleftarrow{\$} \mathbb{Z}_p; \begin{pmatrix} \mathbf{G} \\ x\mathbf{G} \\ x^2\mathbf{G} \\ . \\ . \\ x^q\mathbf{G} \end{pmatrix} \rightarrow \quad \rightarrow x^{q+1}\mathbf{G}$$

Can we reduce DLog to q-DHE?

q-Strong Diffie-Hellman (Boneh Boyen 2004)

- Let $(\mathbb{G}_1, \mathbb{G}_2, e)$ be a bilinear cyclic group of prime order p.

q-Strong Diffie-Hellman (Boneh Boyen 2004)

- Let $(\mathbb{G}_1, \mathbb{G}_2, e)$ be a bilinear cyclic group of prime order p.

$$x \xleftarrow{\$} \mathbb{Z}_p; \begin{pmatrix} \mathbf{G}_1 \\ \mathbf{G}_2 \\ x\mathbf{G}_2 \\ x^2\mathbf{G}_2 \\ . \\ . \\ x^q\mathbf{G}_2 \end{pmatrix} \rightarrow \quad \rightarrow \left(c, \frac{1}{(x+c)}\mathbf{G}_1 \right)$$

q-Strong Diffie-Hellman (Boneh Boyen 2004)

- Let $(\mathbb{G}_1, \mathbb{G}_2, e)$ be a bilinear cyclic group of prime order p.

$$x \xleftarrow{\$} \mathbb{Z}_p; \begin{pmatrix} \mathbf{G}_1 \\ \mathbf{G}_2 \\ x\mathbf{G}_2 \\ x^2\mathbf{G}_2 \\ . \\ . \\ x^q\mathbf{G}_2 \end{pmatrix} \rightarrow \quad \rightarrow \left(c, \frac{1}{(x+c)}\mathbf{G}_1\right)$$

Can we reduce DLog to q-SDH?

DLog

CDH

DHI

one-more DLog

q'-DLog

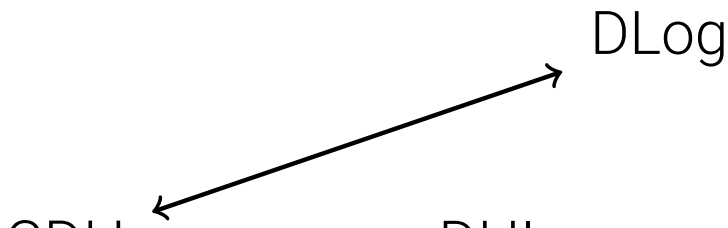
q-SDH

SRDH

Gap-DH

q''-DHE

LRSW



one-more DLog

q'-DLog

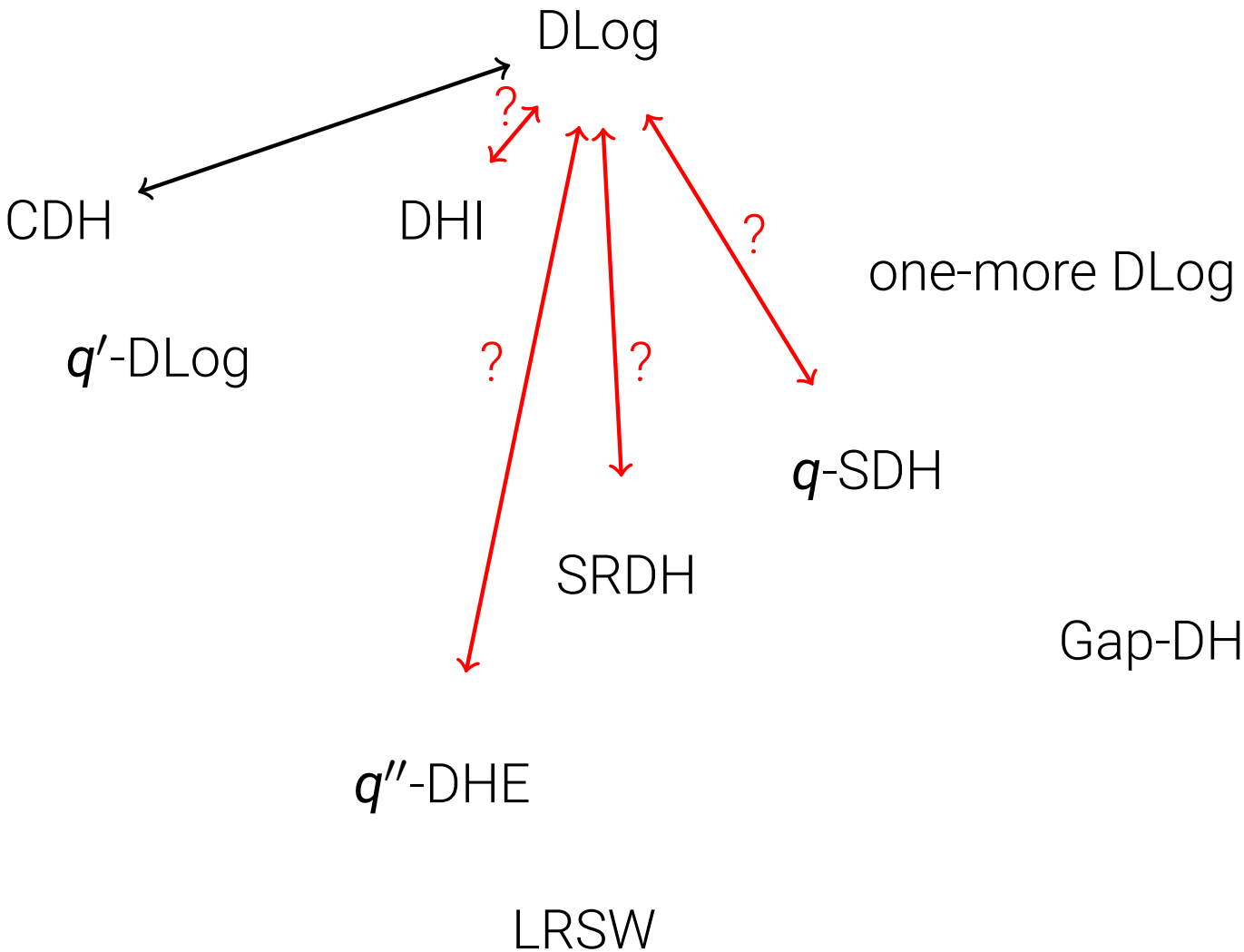
q-SDH

SRDH

Gap-DH

q''-DHE

LRSW


DLog
?
CDH
DHI
?
one-more DLog
q'-DLog
?
?
q-SDH
SRDH
Gap-DH
q''-DHE
LRSW

1. The Algebraic Group Model (FKL 2018)
2. Classification
3. Separation

$(\vec{R}, P)$-uber assumption (Boneh Boyen Goh 2005)

- General idea: Describe many assumptions

$(\vec{R}, P)$-uber assumption (Boneh Boyen Goh 2005)

- General idea: Describe many assumptions
- $\vec{R} \in \mathbb{Z}_p[X_1, \ldots, X_m]^n, P \in \mathbb{Z}_p[X_1, \ldots, X_m]$

$(\vec{R}, P)$-uber assumption (Boneh Boyen Goh 2005)

- General idea: Describe many assumptions
- $\vec{R} \in \mathbb{Z}_p[X_1, \ldots, X_m]^n, P \in \mathbb{Z}_p[X_1, \ldots, X_m]$

$$\vec{x} \xleftarrow{\$} \mathbb{Z}_p^m; \begin{pmatrix} \mathbf{R}_1 = R_1(\vec{x})\mathbf{G} \\ \mathbf{R}_2 = R_2(\vec{x})\mathbf{G} \\ \cdot \\ \cdot \\ \mathbf{R}_n = R_n(\vec{x})\mathbf{G} \end{pmatrix} \rightarrow \quad \rightarrow P(\vec{x})\mathbf{G}$$

$(\vec{R}, P)$-uber assumption (Boneh Boyen Goh 2005)

- General idea: Describe many assumptions (like CDH)
- $\vec{R} \in \mathbb{Z}_p[X_1, \ldots, X_m]^n$, $P \in \mathbb{Z}_p[X_1, \ldots, X_m]$

$$(x, y) \xleftarrow{\$} \mathbb{Z}_p^2; \begin{pmatrix} \mathbf{R}_1 = R_1(\vec{x})\mathbf{G}\ (= 1\mathbf{G}) \\ \mathbf{R}_2 = R_2(\vec{x})\mathbf{G}\ (= x\mathbf{G}) \\ \mathbf{R}_3 = R_3(\vec{x})\mathbf{G}\ (= y\mathbf{G}) \end{pmatrix} \rightarrow \quad \rightarrow P(\vec{x})\mathbf{G}\ (= xy\mathbf{G})$$

$(\vec{R}, P)$-uber assumption (Boneh Boyen Goh 2005)

- General idea: Describe many assumptions (like q-DHE)
- $\vec{R} \in \mathbb{Z}_p[X_1, \ldots, X_m]^n$, $P \in \mathbb{Z}_p[X_1, \ldots, X_m]$

$$x \xleftarrow{\$} \mathbb{Z}_p; \begin{pmatrix} \mathbf{R}_1 = R_1(\vec{x})\mathbf{G}\ (= 1\mathbf{G}) \\ \mathbf{R}_2 = R_2(\vec{x})\mathbf{G}\ (= x\mathbf{G}) \\ \mathbf{R}_3 = R_3(\vec{x})\mathbf{G}\ (= x^2\mathbf{G}) \\ \vdots \\ \mathbf{R}_n = R_n(\vec{x})\mathbf{G}\ (= x^q\mathbf{G}) \end{pmatrix} \rightarrow \quad \rightarrow P(\vec{x})\mathbf{G}\ (= x^{q+1}\mathbf{G})$$

$(\vec{R}, P)$-uber assumption (Boneh Boyen Goh 2005)

- General idea: Describe many assumptions
- $\vec{R} \in \mathbb{Z}_p[X_1, \ldots, X_m]^n, P \in \mathbb{Z}_p[X_1, \ldots, X_m]$

$$\vec{x} \xleftarrow{\$} \mathbb{Z}_p^m; \begin{pmatrix} \mathbf{R}_1 = R_1(\vec{x})\mathbf{G} \\ \mathbf{R}_2 = R_2(\vec{x})\mathbf{G} \\ \cdot \\ \cdot \\ \mathbf{R}_n = R_n(\vec{x})\mathbf{G} \end{pmatrix} \rightarrow \quad \rightarrow P(\vec{x})\mathbf{G}$$

- Easy if $P \in \mathrm{Span}(\vec{R})$:

$(\vec{R}, P)$-uber assumption (Boneh Boyen Goh 2005)

- General idea: Describe many assumptions
- $\vec{R} \in \mathbb{Z}_p[X_1, \ldots, X_m]^n, P \in \mathbb{Z}_p[X_1, \ldots, X_m]$

$$\vec{x} \xleftarrow{\$} \mathbb{Z}_p^m; \begin{pmatrix} \mathbf{R}_1 = R_1(\vec{x})\mathbf{G} \\ \mathbf{R}_2 = R_2(\vec{x})\mathbf{G} \\ \cdot \\ \cdot \\ \mathbf{R}_n = R_n(\vec{x})\mathbf{G} \end{pmatrix} \rightarrow \quad \rightarrow P(\vec{x})\mathbf{G}$$

- Easy if $P \in \mathrm{Span}(\vec{R}) : P = \sum a_i R_i$

$(\vec{R}, P)$-uber assumption (Boneh Boyen Goh 2005)

- General idea: Describe many assumptions
- $\vec{R} \in \mathbb{Z}_p[X_1, \ldots, X_m]^n, P \in \mathbb{Z}_p[X_1, \ldots, X_m]$

$$\vec{x} \xleftarrow{\$} \mathbb{Z}_p^m; \begin{pmatrix} \mathbf{R}_1 = R_1(\vec{x})\mathbf{G} \\ \mathbf{R}_2 = R_2(\vec{x})\mathbf{G} \\ \cdot \\ \cdot \\ \mathbf{R}_n = R_n(\vec{x})\mathbf{G} \end{pmatrix} \rightarrow \quad \rightarrow P(\vec{x})\mathbf{G}$$

- Easy if $P \in \mathrm{Span}(\vec{R}) : P = \sum a_i R_i$

$$P(\vec{x}) = \sum a_i R_i(\vec{x})$$

$(\vec{R}, P)$-uber assumption (Boneh Boyen Goh 2005)

- General idea: Describe many assumptions
- $\vec{R} \in \mathbb{Z}_p[X_1, \ldots, X_m]^n, P \in \mathbb{Z}_p[X_1, \ldots, X_m]$

$$\vec{x} \xleftarrow{\$} \mathbb{Z}_p^m; \begin{pmatrix} \mathbf{R}_1 = R_1(\vec{x})\mathbf{G} \\ \mathbf{R}_2 = R_2(\vec{x})\mathbf{G} \\ \cdot \\ \cdot \\ \mathbf{R}_n = R_n(\vec{x})\mathbf{G} \end{pmatrix} \rightarrow \quad \rightarrow P(\vec{x})\mathbf{G}$$

- Easy if $P \in \mathrm{Span}(\vec{R}) : P = \sum a_i R_i$

$$P(\vec{x}) = \sum a_i R_i(\vec{x})$$
$$P(\vec{x})\mathbf{G} = \sum a_i \mathbf{R}_i$$

$(\vec{R}, P)$-uber assumption (Boneh Boyen Goh 2005)

- General idea: Describe many assumptions
- $\vec{R} \in \mathbb{Z}_p[X_1, \ldots, X_m]^n, P \in \mathbb{Z}_p[X_1, \ldots, X_m]$

$$\vec{x} \xleftarrow{\$} \mathbb{Z}_p^m; \begin{pmatrix} \mathbf{R}_1 = R_1(\vec{x})\mathbf{G} \\ \mathbf{R}_2 = R_2(\vec{x})\mathbf{G} \\ \cdot \\ \cdot \\ \mathbf{R}_n = R_n(\vec{x})\mathbf{G} \end{pmatrix} \rightarrow \quad \rightarrow P(\vec{x})\mathbf{G}$$

- Easy if $P \in \mathrm{Span}(\vec{R})$: $P = \sum a_i R_i$

$$P(\vec{x}) = \sum a_i R_i(\vec{x})$$
$$P(\vec{x})\mathbf{G} = \sum a_i \mathbf{R}_i$$

- Hard in the GGM if $P \notin \mathrm{Span}(\vec{R})$ (non-triviality condition)

q-Strong Diffie-Hellman (Boneh Boyen 2004)

$$x \xleftarrow{\$} \mathbb{Z}_p; \begin{pmatrix} \mathbf{G}_1 \\ \mathbf{G}_2 \\ x\mathbf{G}_2 \\ x^2\mathbf{G}_2 \\ . \\ . \\ x^q\mathbf{G}_2 \end{pmatrix} \rightarrow \quad \rightarrow \left(c, \frac{1}{(x+c)}\mathbf{G}_1\right)$$

q-Strong Diffie-Hellman (Boneh Boyen 2004)

$$x \xleftarrow{\$} \mathbb{Z}_p; \begin{pmatrix} \mathbf{G}_1 \\ \mathbf{G}_2 \\ x\mathbf{G}_2 \\ x^2\mathbf{G}_2 \\ . \\ . \\ x^q\mathbf{G}_2 \end{pmatrix} \rightarrow \quad \rightarrow \left(P \in \left\{ \frac{1}{X+c} \right\}_{c \in \mathbb{Z}_p}, P(x)\mathbf{G}_1 \right)$$

Generalization

- Group $\rightarrow$ Bilinear Group (type $1, 2, 3$)

Generalization

- Group $\rightarrow$ Bilinear Group (type $1, 2, 3$)
- Polynomials $\rightarrow$ Rational fractions

Generalization

- Group $\rightarrow$ Bilinear Group (type $1, 2, 3$)
- Polynomials $\rightarrow$ Rational fractions
- Constant targets $\rightarrow$ Flexible targets

q-DLog

- General idea: Generalize DLog assumption

$$x \xleftarrow{\$} \mathbb{Z}_p; \begin{pmatrix} \mathbf{G} \\ x\mathbf{G} \\ x^2\mathbf{G} \\ . \\ . \\ x^q\mathbf{G} \end{pmatrix} \rightarrow \quad \rightarrow x$$

Univariate case

- We can reduce q-DLog to a non-trivial $(\vec{R}, P)$-uber assumption:

Univariate case

- We can reduce q-DLog to a non-trivial $(\vec{R}, P)$-uber assumption:
- Let $\mathcal{A}$ an adversary against $((R_1, \ldots, R_n), P)$-uber

Univariate case

- We can reduce q-DLog to a non-trivial $(\vec{R}, P)$-uber assumption:
- Let $\mathcal{A}$ an adversary against $((R_1, \ldots, R_n), P)$-uber
- Let q such that $\forall i : \deg(R_i) \leq q$

Univariate case

- We can reduce q-DLog to a non-trivial $(\vec{R}, P)$-uber assumption:
- Let $\mathcal{A}$ an adversary against $((R_1, \ldots, R_n), P)$-uber
- Let q such that $\forall i : \deg(R_i) \leq q$
- Let's break q-DLog

Univariate case

- We can reduce q-DLog to a non-trivial $(\vec{R}, P)$-uber assumption:
- Let $\mathcal{A}$ an adversary against $((R_1, \ldots, R_n), P)$-uber
- Let q such that $\forall i : \deg(R_i) \leq q$
- Let's break q-DLog
- $\mathcal{B}^{\mathcal{A}}(\mathbf{X}^{(0)}, \mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(q)})$:

Univariate case

- We can reduce q-DLog to a non-trivial $(\vec{R}, P)$-uber assumption:
- Let $\mathcal{A}$ an adversary against $((R_1, \ldots, R_n), P)$-uber
- Let q such that $\forall i : \deg(R_i) \leq q$
- Let's break q-DLog
- $\mathcal{B}^{\mathcal{A}}(\mathbf{X}^{(0)}, \mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(q)})$:
  - $\mathbf{R}_i := \sum r_{i,j} \mathbf{X}^{(j)}$

Univariate case

- We can reduce q-DLog to a non-trivial $(\vec{R}, P)$-uber assumption:
- Let $\mathcal{A}$ an adversary against $((R_1, \ldots, R_n), P)$-uber
- Let q such that $\forall i : \deg(R_i) \leq q$
- Let's break q-DLog
- $\mathcal{B}^{\mathcal{A}}(\mathbf{X}^{(0)}, \mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(q)})$:
  - $\mathbf{R}_i := \sum r_{i,j} \mathbf{X}^{(j)} = \sum R_i(x)\mathbf{G}$

Univariate case

- We can reduce q-DLog to a non-trivial $(\vec{R}, P)$-uber assumption:
- Let $\mathcal{A}$ an adversary against $((R_1, \ldots, R_n), P)$-uber
- Let q such that $\forall i : \deg(R_i) \leq q$
- Let's break q-DLog
- $\mathcal{B}^{\mathcal{A}}(\mathbf{X}^{(0)}, \mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(q)})$:
  - $\mathbf{R}_i := \sum r_{i,j} \mathbf{X}^{(j)} = \sum R_i(x)\mathbf{G}$
  - $(\mathbf{P}, a_1, \ldots, a_n) \leftarrow \mathcal{A}(\mathbf{R}_1, \ldots, \mathbf{R}_n)$

Univariate case

- We can reduce q-DLog to a non-trivial $(\vec{R}, P)$-uber assumption:
- Let $\mathcal{A}$ an adversary against $((R_1, \ldots, R_n), P)$-uber
- Let q such that $\forall i : \deg(R_i) \leq q$
- Let's break q-DLog
- $\mathcal{B}^{\mathcal{A}}(\mathbf{X}^{(0)}, \mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(q)})$:
  - $\mathbf{R}_i := \sum r_{i,j} \mathbf{X}^{(j)} = \sum R_i(x)\mathbf{G}$
  - $(\mathbf{P}, a_1, \ldots, a_n) \leftarrow \mathcal{A}(\mathbf{R}_1, \ldots, \mathbf{R}_n)$
    $(\sum a_i \mathbf{R}_i = \mathbf{P})$

Univariate case

- We can reduce q-DLog to a non-trivial $(\vec{R}, P)$-uber assumption:
- Let $\mathcal{A}$ an adversary against $((R_1, \ldots, R_n), P)$-uber
- Let q such that $\forall i : \deg(R_i) \leq q$
- Let's break q-DLog
- $\mathcal{B}^{\mathcal{A}}(\mathbf{X}^{(0)}, \mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(q)})$:
  - $\mathbf{R}_i := \sum r_{i,j} \mathbf{X}^{(j)} = \sum R_i(x)\mathbf{G}$
  - $(\mathbf{P}, a_1, \ldots, a_n) \leftarrow \mathcal{A}(\mathbf{R}_1, \ldots, \mathbf{R}_n)$ $(\sum a_i \mathbf{R}_i = \mathbf{P})$
  - $\{x_1^*, \ldots, x_q^*\} \leftarrow$ Solve $(\sum a_i R_i(X) = P(X))$

Univariate case

- We can reduce q-DLog to a non-trivial $(\vec{R}, P)$-uber assumption:
- Let $\mathcal{A}$ an adversary against $((R_1, \ldots, R_n), P)$-uber
- Let q such that $\forall i : \deg(R_i) \leq q$
- Let's break q-DLog
- $\mathcal{B}^{\mathcal{A}}(\mathbf{X}^{(0)}, \mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(q)})$:
  - $\mathbf{R}_i := \sum r_{i,j} \mathbf{X}^{(j)} = \sum R_i(x)\mathbf{G}$
  - $(\mathbf{P}, a_1, \ldots, a_n) \leftarrow \mathcal{A}(\mathbf{R}_1, \ldots, \mathbf{R}_n)$ $(\sum a_i \mathbf{R}_i = \mathbf{P})$
  - $\{x_1^*, \ldots, x_q^*\} \leftarrow$ Solve $(\sum a_i R_i(X) = P(X))$
  - Output x_i^* such that $x_i^* \mathbf{G} = \mathbf{X}^{(1)}$

Generalization

- **Uber**:
  - Group $\rightarrow$ Bilinear Group (type $1, 2, 3$)

Generalization

- **Uber**:
  - Group $\rightarrow$ Bilinear Group (type $1, 2, 3$)
  - Univariate $\rightarrow$ Multivariate (CDH)
    (embed the challenge in every coordinate: $x_i := y_i x + v_i$)

Generalization

- **Uber**:
  - Group $\rightarrow$ Bilinear Group (type $1, 2, 3$)
  - Univariate $\rightarrow$ Multivariate (CDH)
    (embed the challenge in every coordinate: $x_i := y_i x + v_i$)
  - Fixed targets $\rightarrow$ Flexible targets

Generalization

- **Uber**:
  - Group $\rightarrow$ Bilinear Group (type $1, 2, 3$)
  - Univariate $\rightarrow$ Multivariate (CDH)
    (embed the challenge in every coordinate: $x_i := y_i x + v_i$)
  - Fixed targets $\rightarrow$ Flexible targets
    $\mathcal{A}$ can choose $P \notin \mathrm{Span}(R)$.

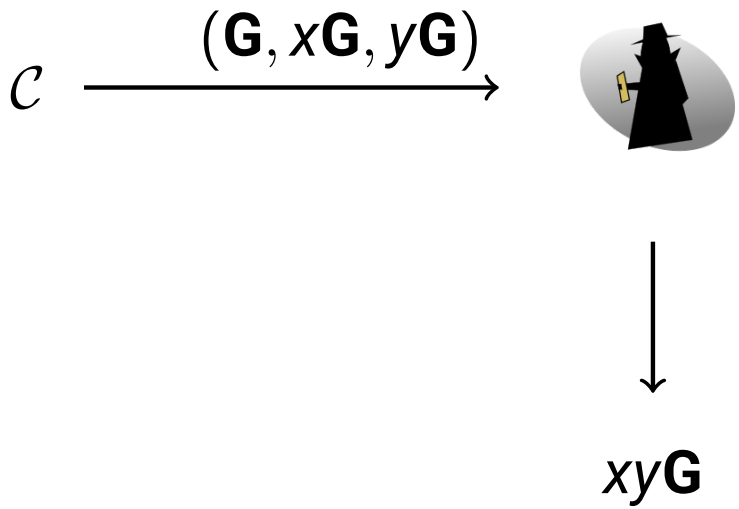
Generalization

- **Uber**:
  - Group $\rightarrow$ Bilinear Group (type $1, 2, 3$)
  - Univariate $\rightarrow$ Multivariate (CDH)
    (embed the challenge in every coordinate: $x_i := y_i x + v_i$)
  - Fixed targets $\rightarrow$ Flexible targets
    $\mathcal{A}$ can choose $P \notin \mathrm{Span}(R)$.
- **Ruber**: Polynomials $\rightarrow$ Rational fractions (q-SDH)

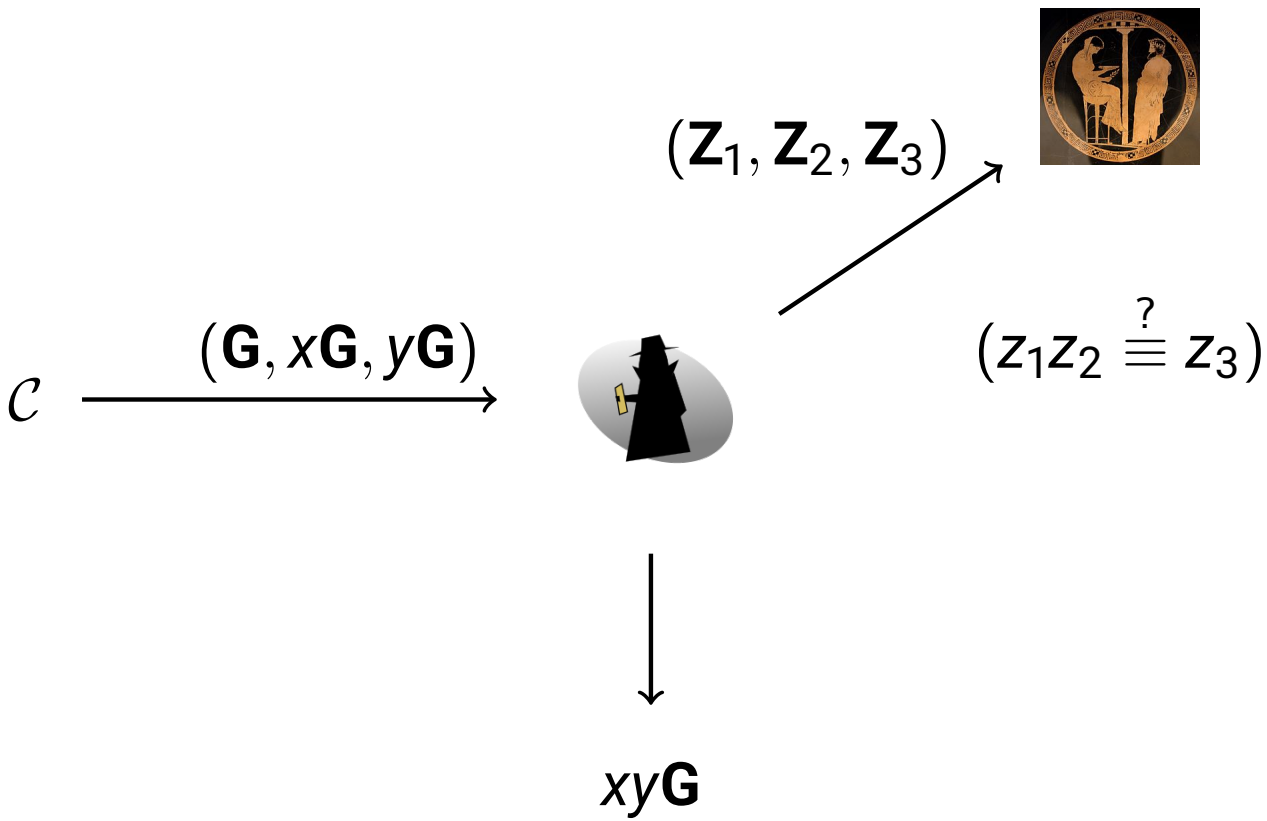
Gap-DH

$\mathcal{C}$ $\xrightarrow{(\mathbf{G}, x\mathbf{G}, y\mathbf{G})}$ 

Gap-DH



Gap-DH


$(\mathbf{Z}_1, \mathbf{Z}_2, \mathbf{Z}_3)$
$\mathcal{C}$
$(\mathbf{G}, x\mathbf{G}, y\mathbf{G})$
$(z_1 z_2 \stackrel{?}{\equiv} z_3)$
$xy\mathbf{G}$

Generalization

- **Uber**:
  - Group $\rightarrow$ Bilinear Group (types $1, 2, 3$)
  - Univariate $\rightarrow$ Multivariate (CDH)
    (embed the challenge in every coordinate: $x_i := y_i z + v_i$)
  - Fixed targets $\rightarrow$ Flexible targets
    $\mathcal{A}$ can choose $P \notin \mathrm{Span}(R)$
- **Ruber**: Polynomials $\rightarrow$ Rational fractions (q-SDH)

Generalization

- **Uber**:
  - Group $\rightarrow$ Bilinear Group (types $1, 2, 3$)
  - Univariate $\rightarrow$ Multivariate (CDH)
    (embed the challenge in every coordinate: $x_i := y_i z + v_i$)
  - Fixed targets $\rightarrow$ Flexible targets
    $\mathcal{A}$ can choose $P \notin \mathrm{Span}(R)$
- **Ruber**: Polynomials $\rightarrow$ Rational fractions (q-SDH)
- **Druber**: Add decisional oracles (Gap-DH) **(New)**

$$\rightarrow \begin{pmatrix} m^* \\ a^*\mathbf{G} \\ a^*(x + m^*xy)\mathbf{G} \end{pmatrix}$$

Generalization

- **Uber**:
  - Group $\rightarrow$ Bilinear Group (types $1, 2, 3$)
  - Univariate $\rightarrow$ Multivariate (CDH)
    (embed the challenge in every coordinate: $x_i := y_i z + v_i$)
  - Fixed targets $\rightarrow$ Flexible targets
    $\mathcal{A}$ can choose $P \notin \mathrm{Span}(R)$
- **Ruber**: Polynomials $\rightarrow$ Rational fractions (q-SDH)
- **Druber**: Add decisional oracles (Gap-DH) **(New)**

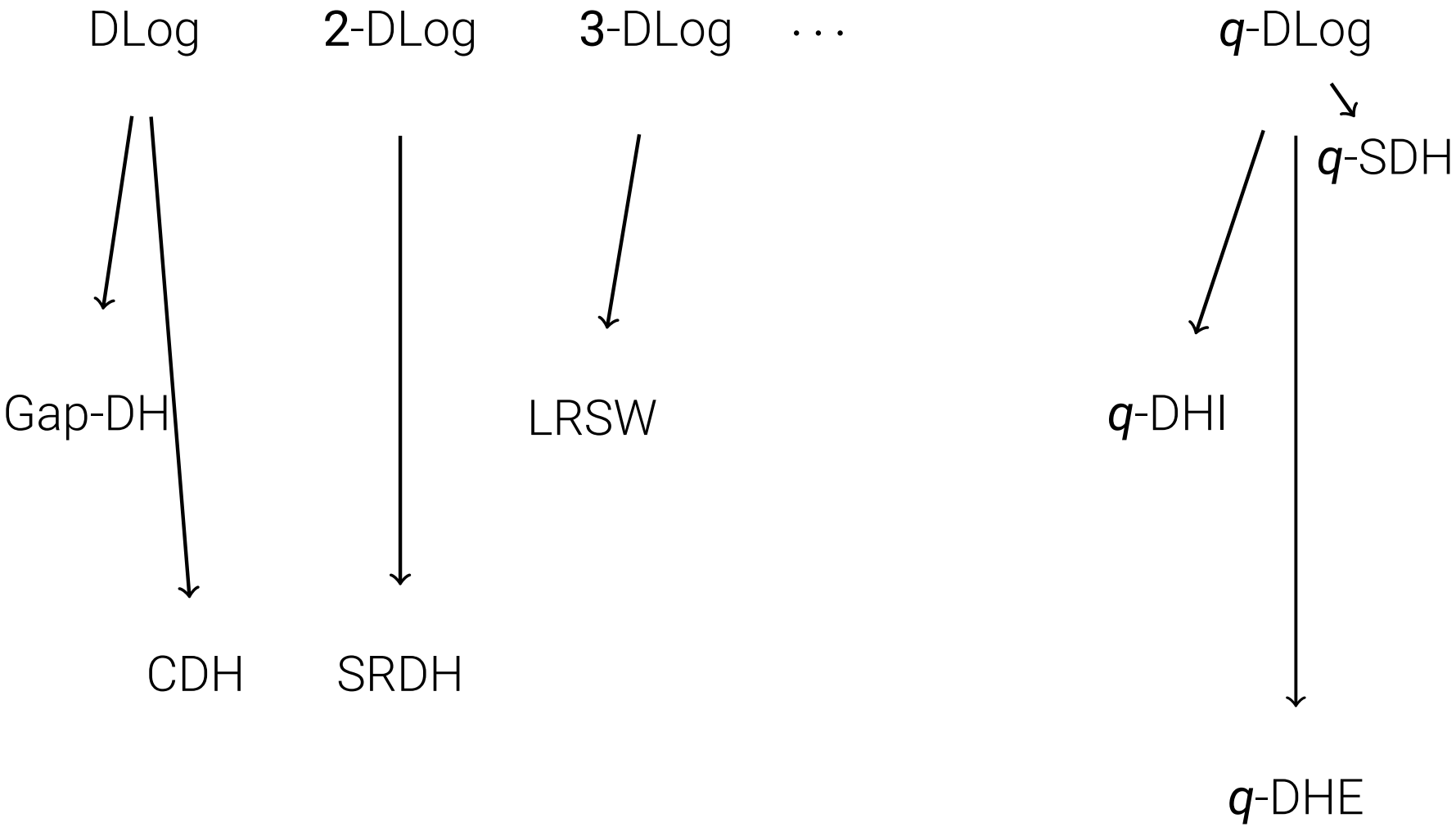
Generalization

- **Uber**:
  - Group $\rightarrow$ Bilinear Group (types $1, 2, 3$)
  - Univariate $\rightarrow$ Multivariate (CDH)
    (embed the challenge in every coordinate: $x_i := y_i z + v_i$)
  - Fixed targets $\rightarrow$ Flexible targets
    $\mathcal{A}$ can choose $P \notin \mathrm{Span}(R)$
- **Ruber**: Polynomials $\rightarrow$ Rational fractions (q-SDH)
- **Druber**: Add decisional oracles (Gap-DH) **(New)**
- **Gegenuber**: Constant generator $\rightarrow$ generate its own generator (LRSW) **(New)**

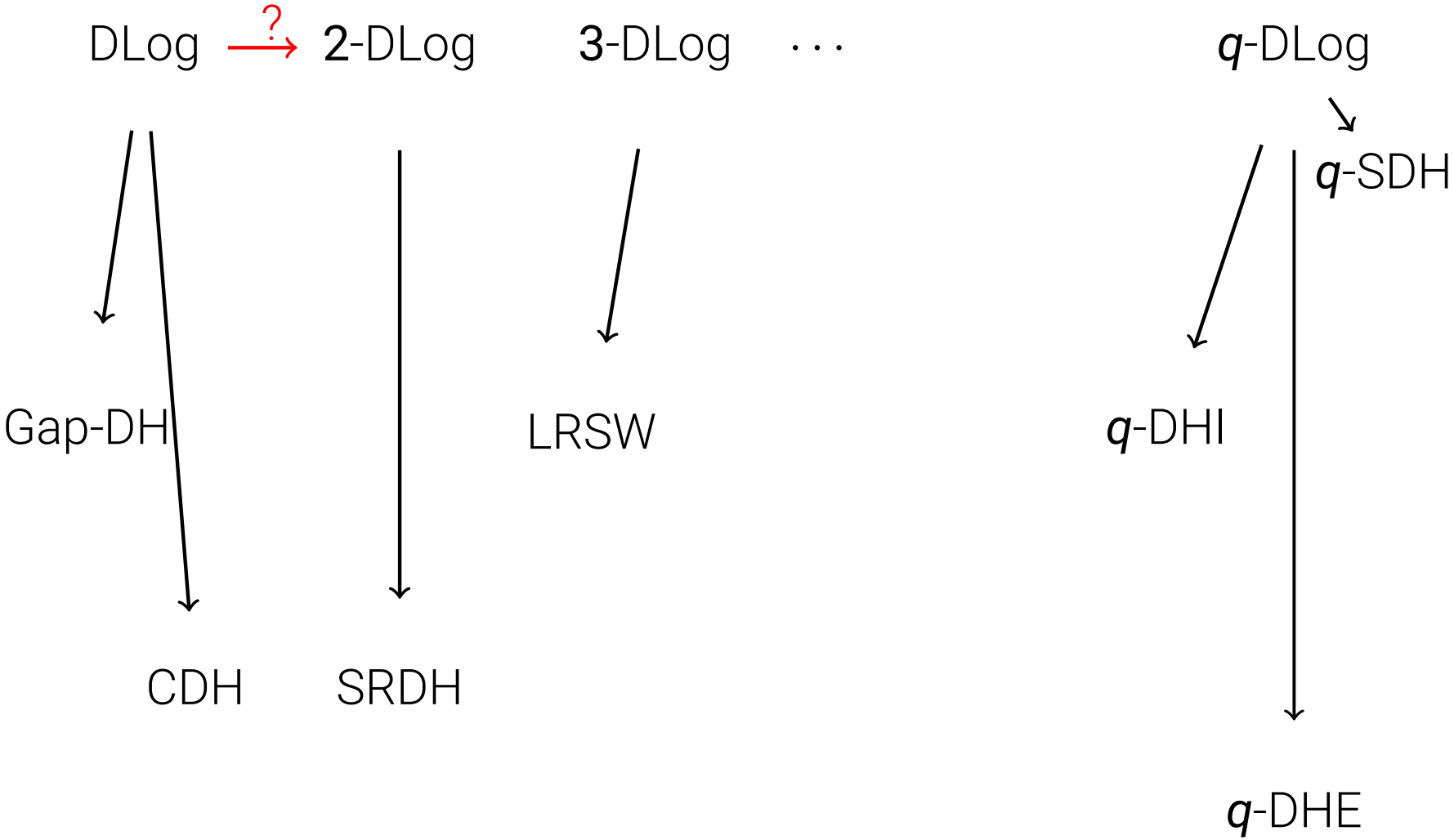
Generalization

- **Uber**:
  - Group $\rightarrow$ Bilinear Group (types $1, 2, 3$)
  - Univariate $\rightarrow$ Multivariate (CDH)
    (embed the challenge in every coordinate: $x_i := y_i z + v_i$)
  - Fixed targets $\rightarrow$ Flexible targets
    $\mathcal{A}$ can choose $P \notin \mathrm{Span}(R)$
- **Ruber**: Polynomials $\rightarrow$ Rational fractions (q-SDH)
- **Druber**: Add decisional oracles (Gap-DH) **(New)**
- **Gegenuber**: Constant generator $\rightarrow$ generate its own generator (LRSW) **(New)**

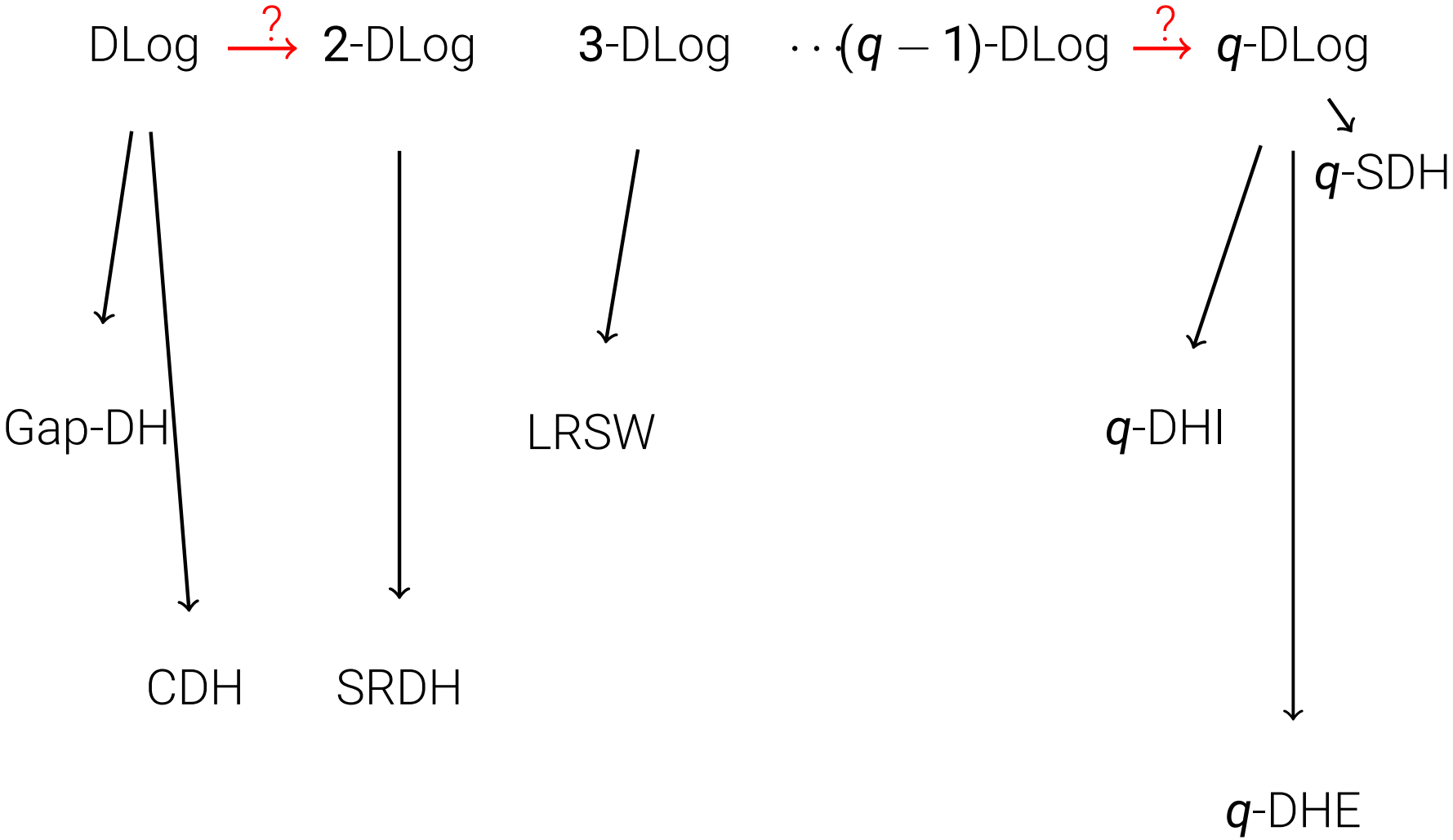
$\mathcal{A}$ can choose $\mathbf{G}'$ and return $(\mathbf{G}', P(\vec{x})\mathbf{G}')$.



Can we do better?

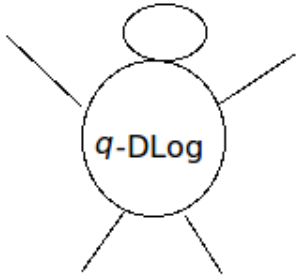


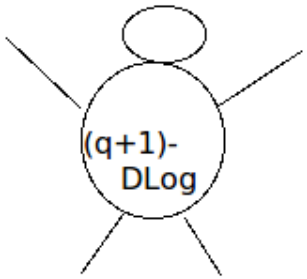
Can we do better?

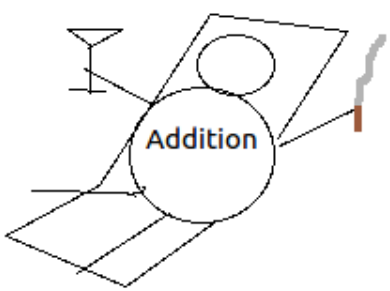


Can we do better?

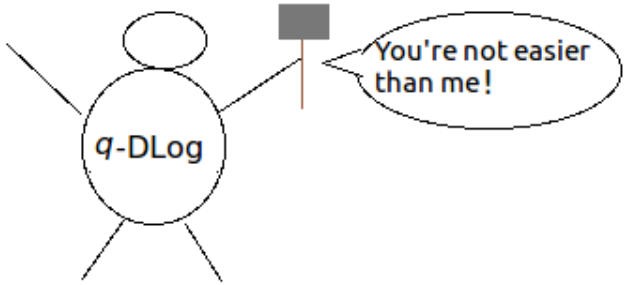
1. The Algebraic Group Model (FKL 2018)
2. Classification
3. Separation

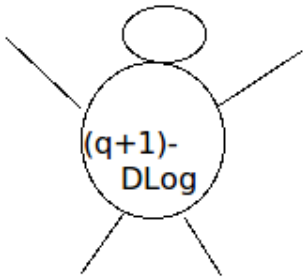


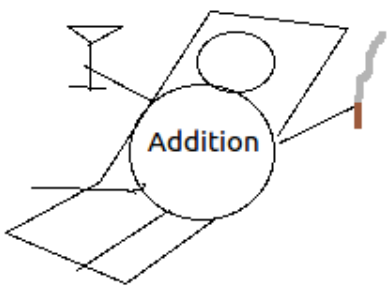




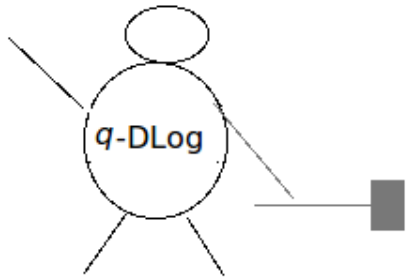
hardness

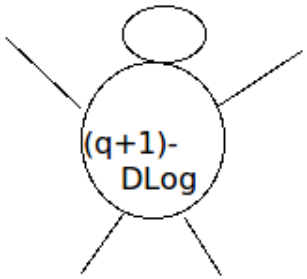


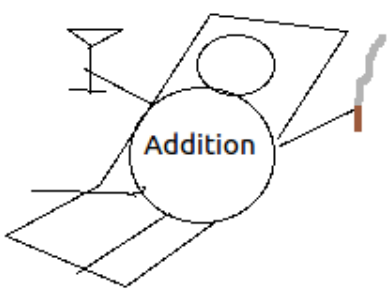




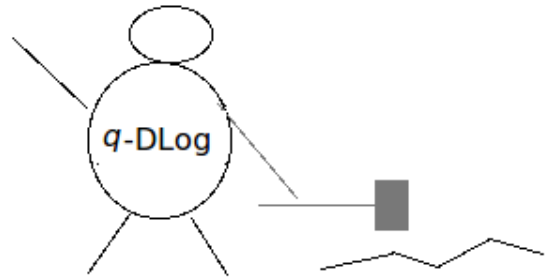
hardness

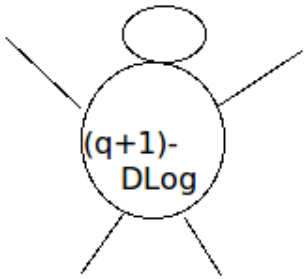


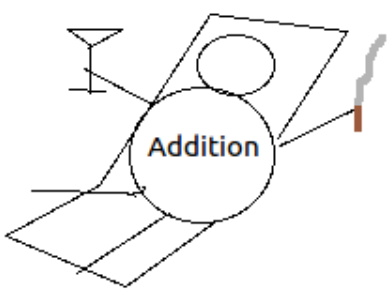




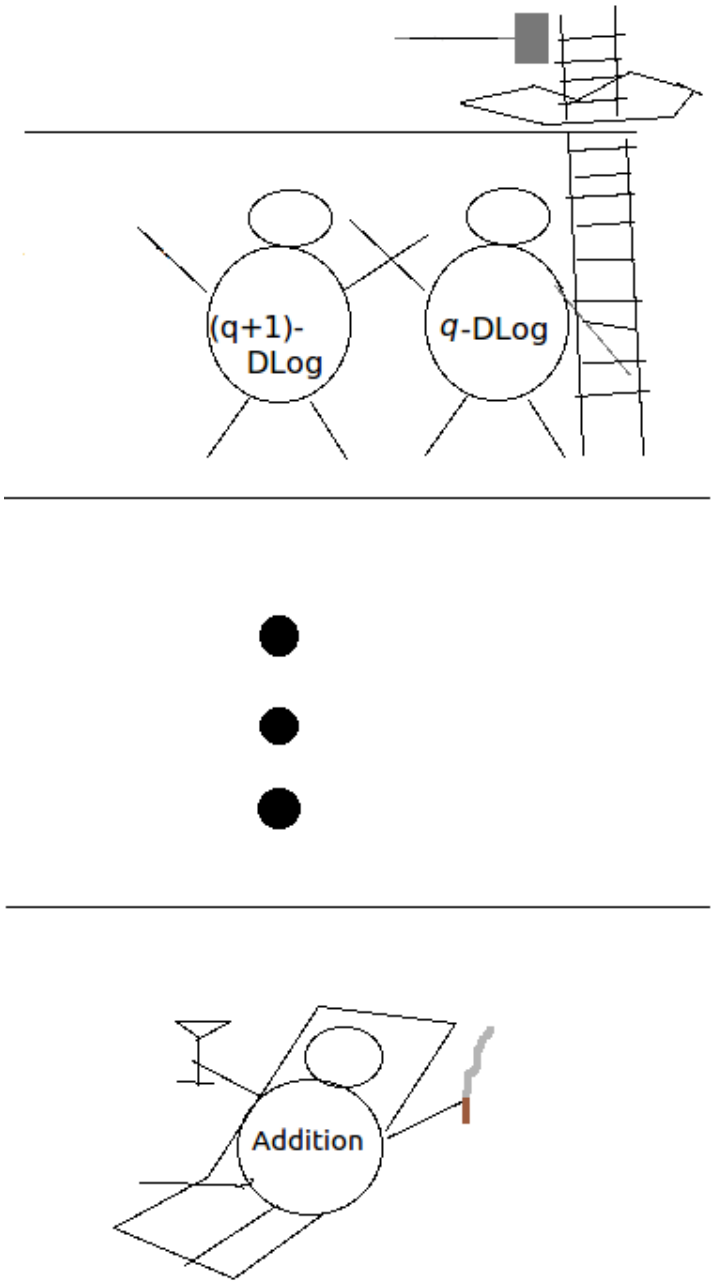
hardness







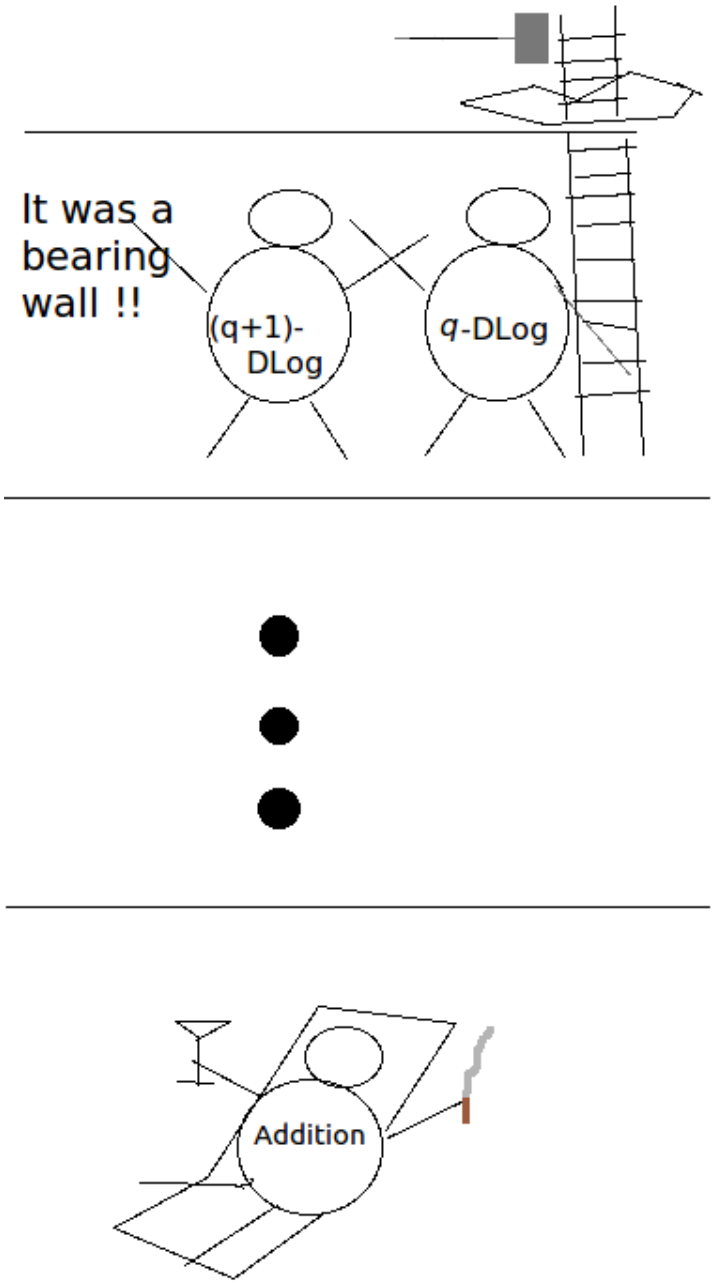
hardness



hardness

Thm:

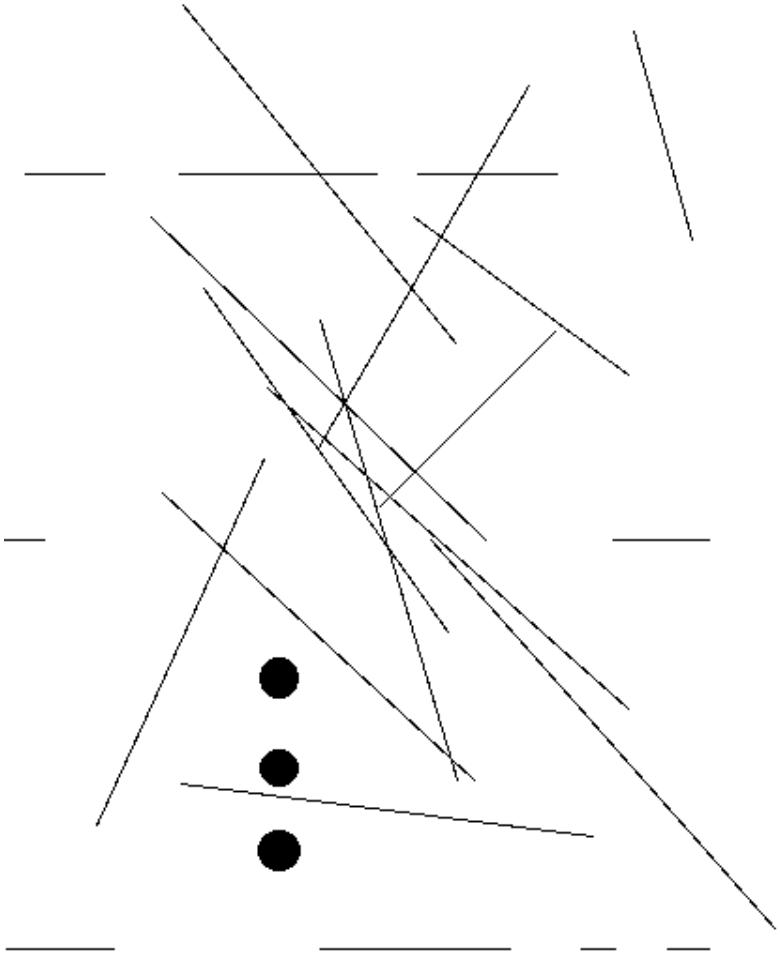
If $(q+1)$-DLog is q-DLog-hard.



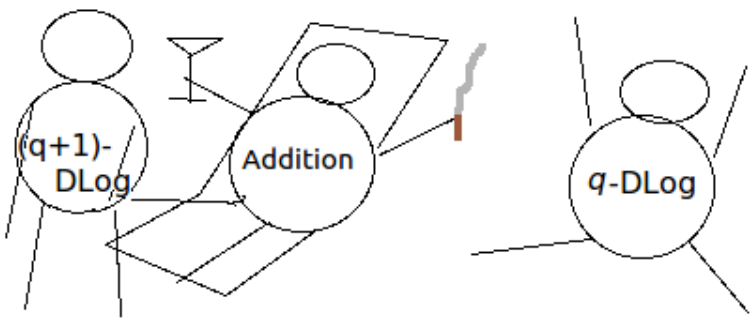
hardness

Thm:

If $(q+1)$-DLog is q-DLog-hard.



hardness

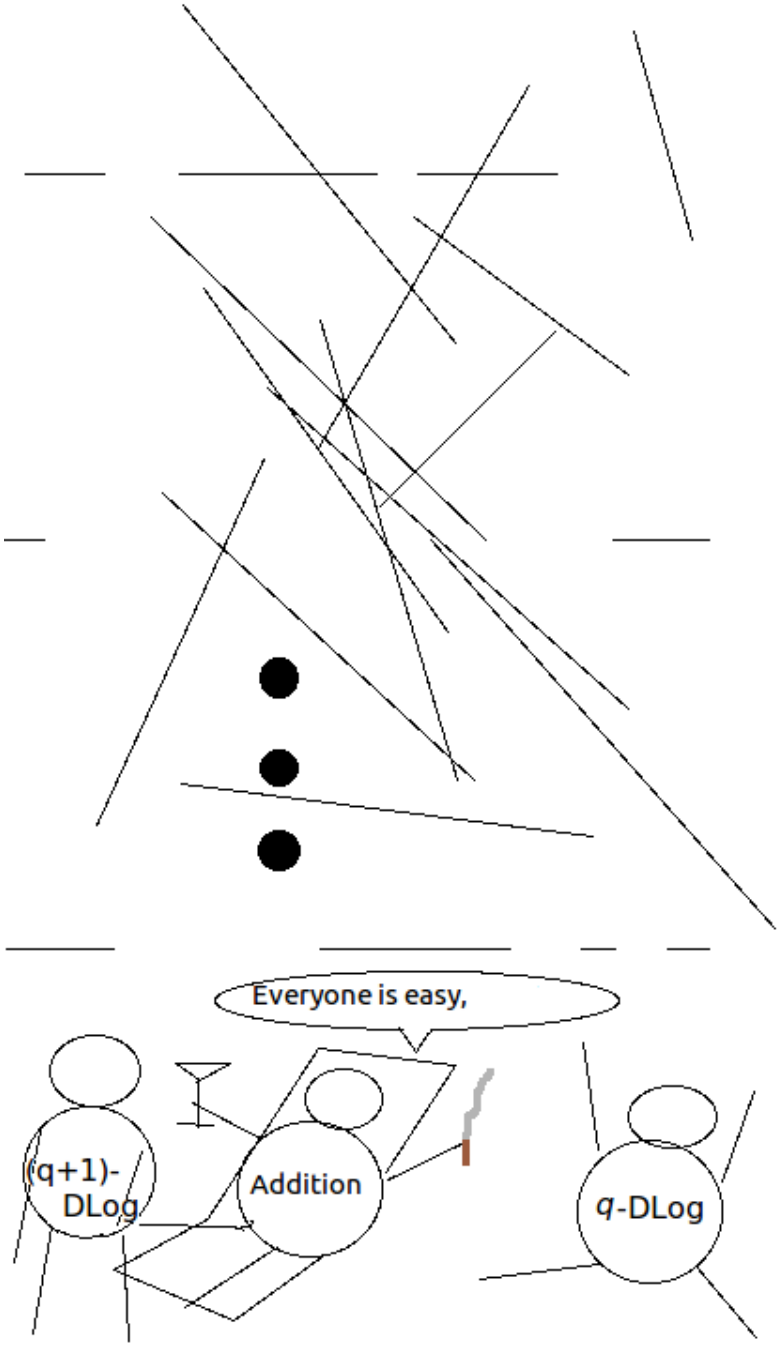


Thm:

If $(q+1)$-DLog
is q-DLog-hard.

↓

q-DLog $\in$ FBPP.



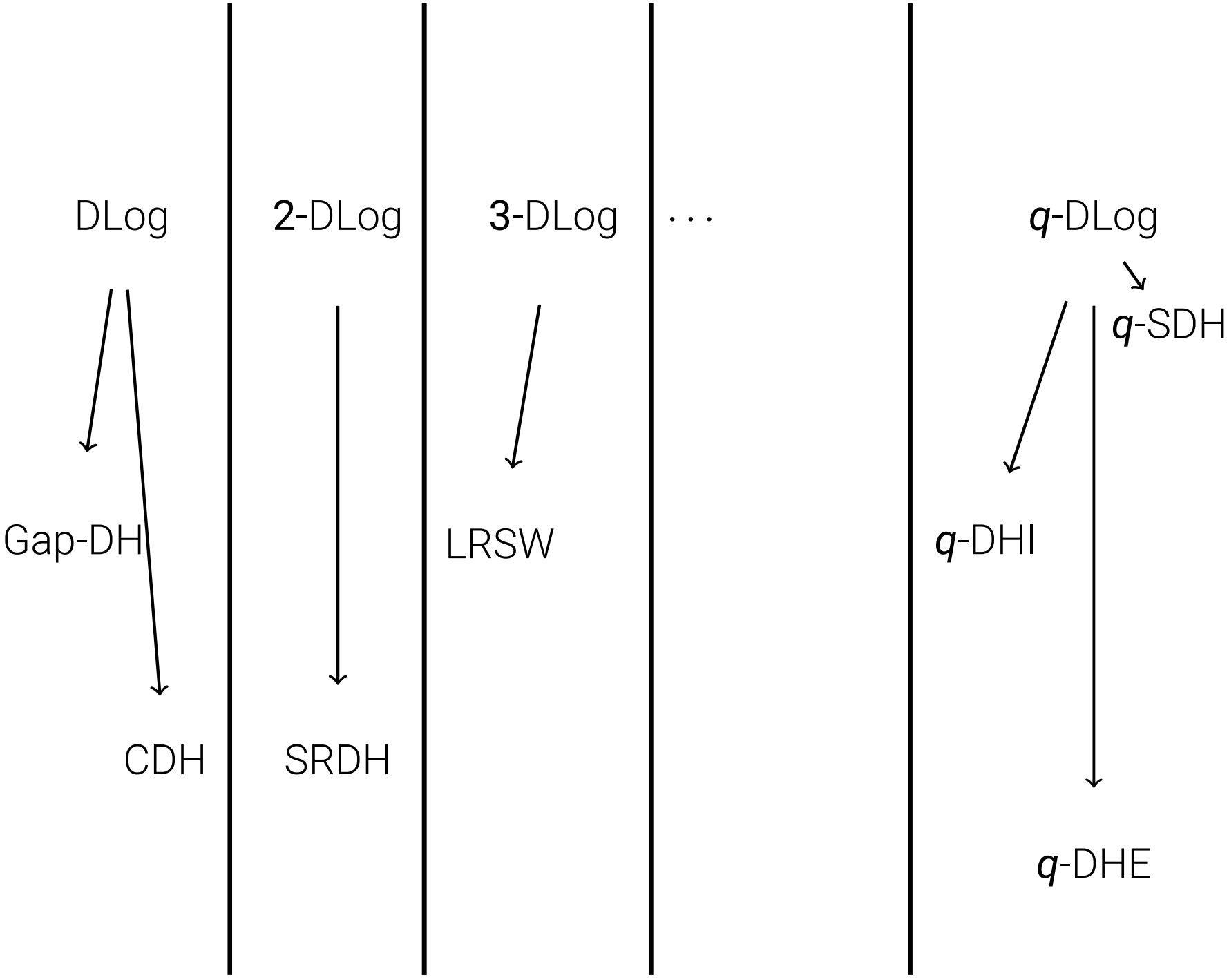
hardness

Thm:

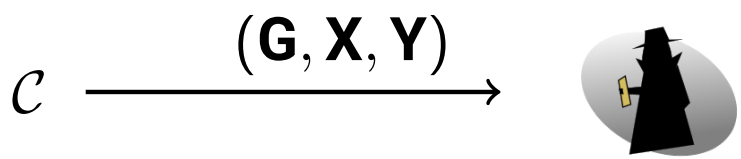
If $(q+1)$-DLog is q-DLog-hard.

↓

q-DLog $\in$ FBPP.


DLog
2-DLog
3-DLog
· · ·
q-DLog
q-SDH
Gap-DH
LRSW
q-DHI
CDH
SRDH
q-DHE

One-More Discrete Log



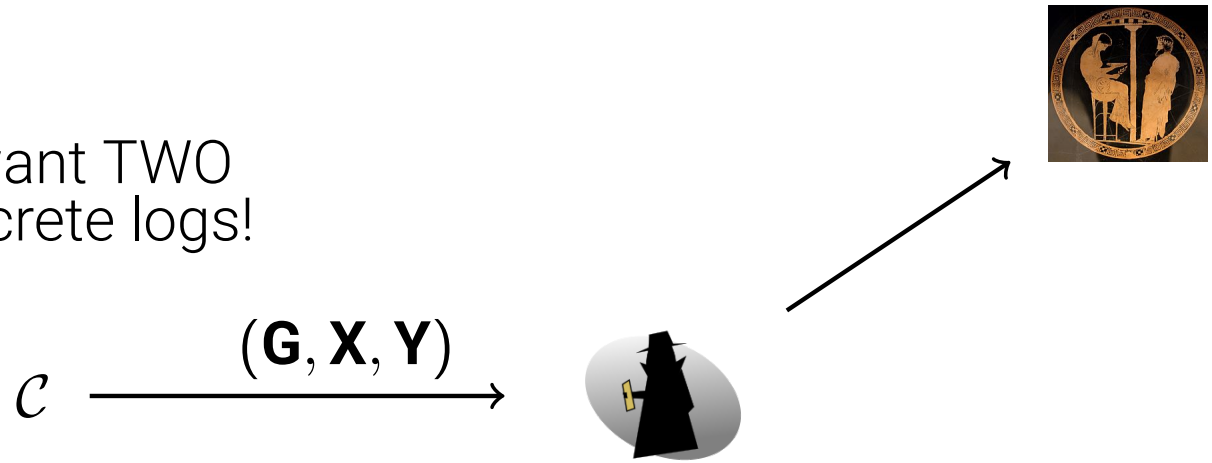
One-More Discrete Log

I want TWO discrete logs!

$\mathcal{C} \xrightarrow{(\mathbf{G}, \mathbf{X}, \mathbf{Y})}$ 

One-More Discrete Log

I want TWO discrete logs!



One-More Discrete Log

I want TWO discrete logs!

$\mathcal{C}$ $\xrightarrow{(\mathbf{G}, \mathbf{X}, \mathbf{Y})}$

I will compute only ONE discrete log for you...

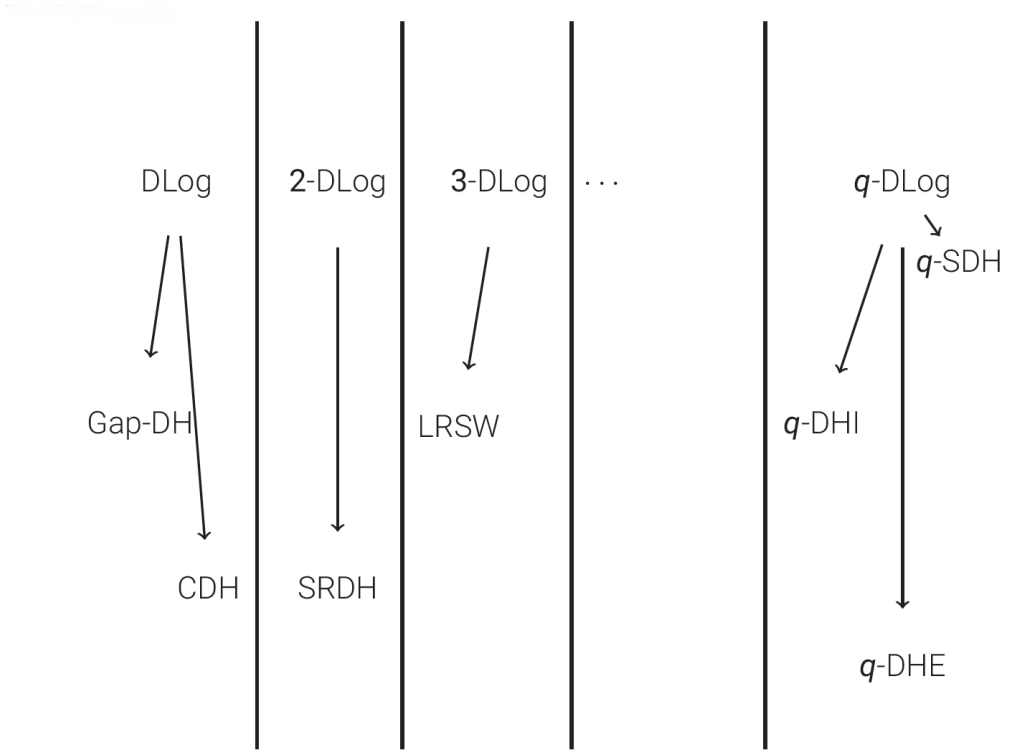
One-More Discrete Log

I want TWO discrete logs!

$\mathcal{C}$ $\xrightarrow{(\mathbf{G}, \mathbf{X}, \mathbf{Y})}$

Life is hard...

I will compute only ONE discrete log for you...



41

OMDL

Thm: *q*-DLog does not imply One-More DLog in the AGM.

Thank you for your attention.