

The Distinction Between Fixed and Random Generators in Group-Based Assumptions

James Bartusek

(Princeton → UC Berkeley)

Fermi Ma

(Princeton)

Mark Zhandry

(Princeton + NTT Research)

The Decisional Diffie-Hellman (DDH) Assumption

Fix a cyclic group G of order q .

Let g be a generator of G .

For uniformly random $x, y, z \leftarrow \mathbb{Z}_q$,

$$(g, g^x, g^y, g^{xy}) \approx_c (g, g^x, g^y, g^z).$$

The Decisional Diffie-Hellman (DDH) Assumption

Fix a cyclic group G of order q .

Let g be a generator of G .

For uniformly random $x, y, z \leftarrow \mathbb{Z}_q$,

$$(g, g^x, g^y, g^{xy}) \approx_c (g, g^x, g^y, g^z).$$

When is g chosen?

The Decisional Diffie-Hellman (DDH) Assumption

Fix a cyclic group G of order q with fixed generator g .

For uniformly random $x, y, z \leftarrow \mathbb{Z}_q$,

$$(g, g^x, g^y, g^{xy}) \approx_c (g, g^x, g^y, g^z).$$

- Katz-Lindell (textbook)
- Boneh (1998 DDH survey)
- Katz-Wang (CCS 2003)
- Boyle-Gilboa-Ishai (CRYPTO 2016)
- Döttling-Garg (CRYPTO 2017)
- Villar (PKC 2017)

g is fixed in the
group description

The Decisional Diffie-Hellman (DDH) Assumption

Fix a cyclic group G of order q with fixed generator g .

Pick a uniformly random $r \leftarrow \mathbb{Z}_q$ and set $h = g^r$.

For uniformly random $x, y, z \leftarrow \mathbb{Z}_q$,

$$(h, h^x, h^y, h^{xy}) \approx_c (h, h^x, h^y, h^z).$$

h is a random
group generator

The Decisional Diffie-Hellman (DDH) Assumption

Fix a cyclic group G of order q with fixed generator g .

Pick a uniformly random $r \leftarrow \mathbb{Z}_q$ and set $h = g^r$.

For uniformly random $x, y, z \leftarrow \mathbb{Z}_q$,

$$(h, h^x, h^y, h^{xy}) \approx_C (h, h^x, h^y, h^z).$$

- Naor-Reingold (FOCS 1995)
- Naor-Reingold (FOCS 1997)
- Cramer-Shoup (CRYPTO 1998)
- Nielsen (CRYPTO 2002)
- Agrawal-Libert-Stehlé (CRYPTO 2016)

h is a random
group generator

(fixed-DDH) For fixed generator g ,
 $(g, g^x, g^y, g^{xy}) \approx_c (g, g^x, g^y, g^z)$.

(random-DDH) For random generator h ,
 $(h, h^x, h^y, h^{xy}) \approx_c (h, h^x, h^y, h^z)$.

Are these assumptions equivalent?

(fixed-DDH) For fixed generator g ,
 $(g, g^x, g^y, g^{xy}) \approx_c (g, g^x, g^y, g^z)$.

(random-DDH) For random generator h ,
 $(h, h^x, h^y, h^{xy}) \approx_c (h, h^x, h^y, h^z)$.

Are these assumptions equivalent?

[Shoup99]: fixed- and random-DDH not known to be equivalent

(also discussed in [SadeghiSteiner01] and [Galbraith] textbook)

(fixed-DDH) For fixed generator g ,
 $(g, g^x, g^y, g^{xy}) \approx_c (g, g^x, g^y, g^z)$.

(random-DDH) For random generator h ,
 $(h, h^x, h^y, h^{xy}) \approx_c (h, h^x, h^y, h^z)$.

Are these assumptions equivalent?

Follow-up question:

Do we have similar issues for Discrete Log or CDH?

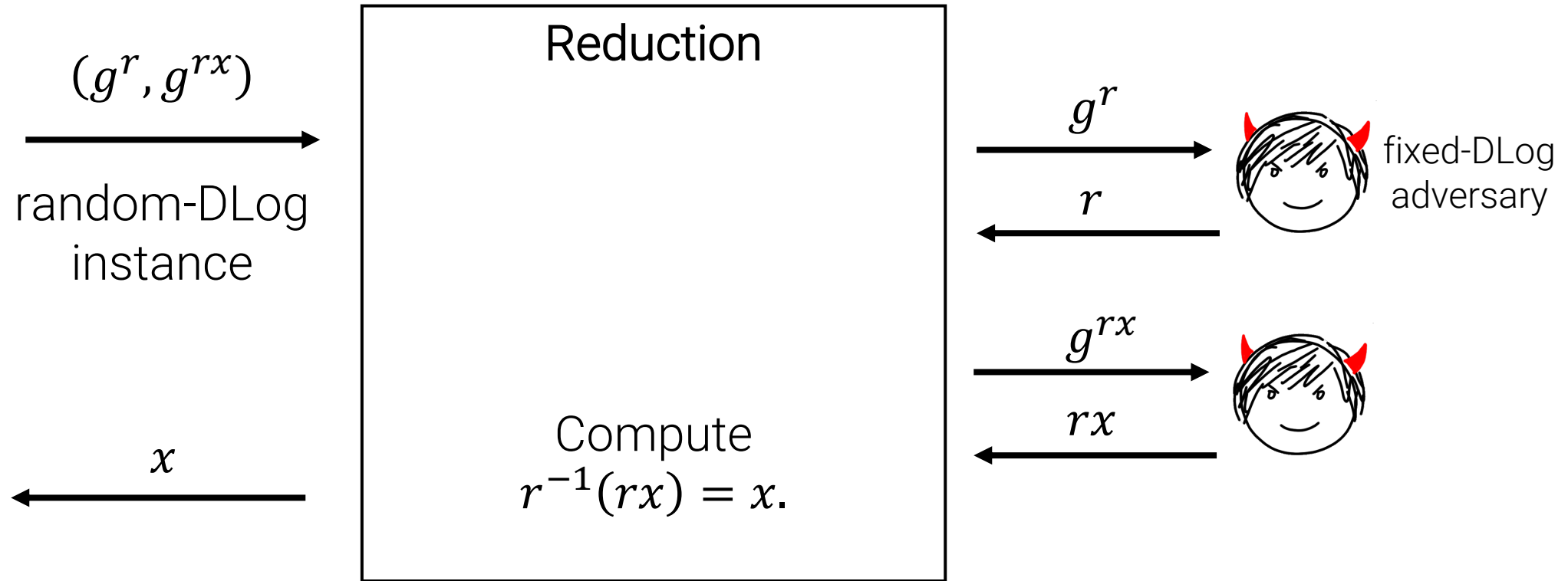
When are fixed and random-generator assumptions equivalent?

Discrete Log	CDH	DDH
equivalent (folklore)	equivalent (folklore)	no known equivalence or separations

Note: Adversary for random-generator problem always implies adversary for fixed-generator problem (re-randomize the fixed-generator instance).

Warmup (folklore): random-DLog \leq_R fixed-DLog.

Public G , prime order p ,
generator g .



When are fixed and random-generator assumptions equivalent?

Discrete Log	CDH	DDH
equivalent (folklore)	equivalent (folklore)	no known equivalence or separations



Folklore CDH equivalence requires knowing totient of group order

When are fixed and random-generator assumptions equivalent?

	Discrete Log	CDH	DDH
known prime order	equivalent (folklore)	equivalent (folklore)	??
unknown prime order	equivalent (folklore)	??	??
unknown factorization	equivalent (folklore)	??	??

	Discrete Log	CDH	DDH
known prime order	equivalent (folklore)	equivalent (folklore)	black-box separated (this work)
unknown order	equivalent (folklore)	black-box separated* (this work)	black-box separated (this work)
unknown factorization	equivalent (folklore)	black-box separated** (this work)	black-box separated (this work)

* Requires hardness of factoring unbalanced modulus

** Requires strong knowledge assumption


Strategy: Prove hardness of random-CDH (resp. DDH) in the generic group model even given an oracle which solves fixed-CDH (resp. DDH).

	Discrete Log	CDH	DDH
known prime order	equivalent (folklore)	equivalent (folklore)	black-box separated (this work)
unknown order	equivalent (folklore)	black-box separated* (this work)	black-box separated (this work)
unknown factorization	equivalent (folklore)	black-box separated** (this work)	black-box separated (this work)

* Requires hardness of factoring unbalanced modulus

** Requires strong knowledge assumption

What if we had concrete groups realizing these separations?



Observation: A group where fixed-CDH is easy but random-CDH is hard implies a “self-bilinear map” [YYHK14].

Self-Bilinear Map: A group G with a pairing $e: G^2 \rightarrow G$ such that

$$e(g^x, g^y) = e(g, g)^{xy}.$$

[YYHK14]: These imply

- multiparty non-interactive key agreement with trusted setup [BS02]
- distributed broadcast encryption [BZ14]

Observation: A group where fixed-CDH is easy but random-CDH is hard implies a “self-bilinear map” [YYHK14].

Self-Bilinear Map: A group G with a pairing $e: G^2 \rightarrow G$ such that

$$e(g^x, g^y) = e(g, g)^{xy}.$$

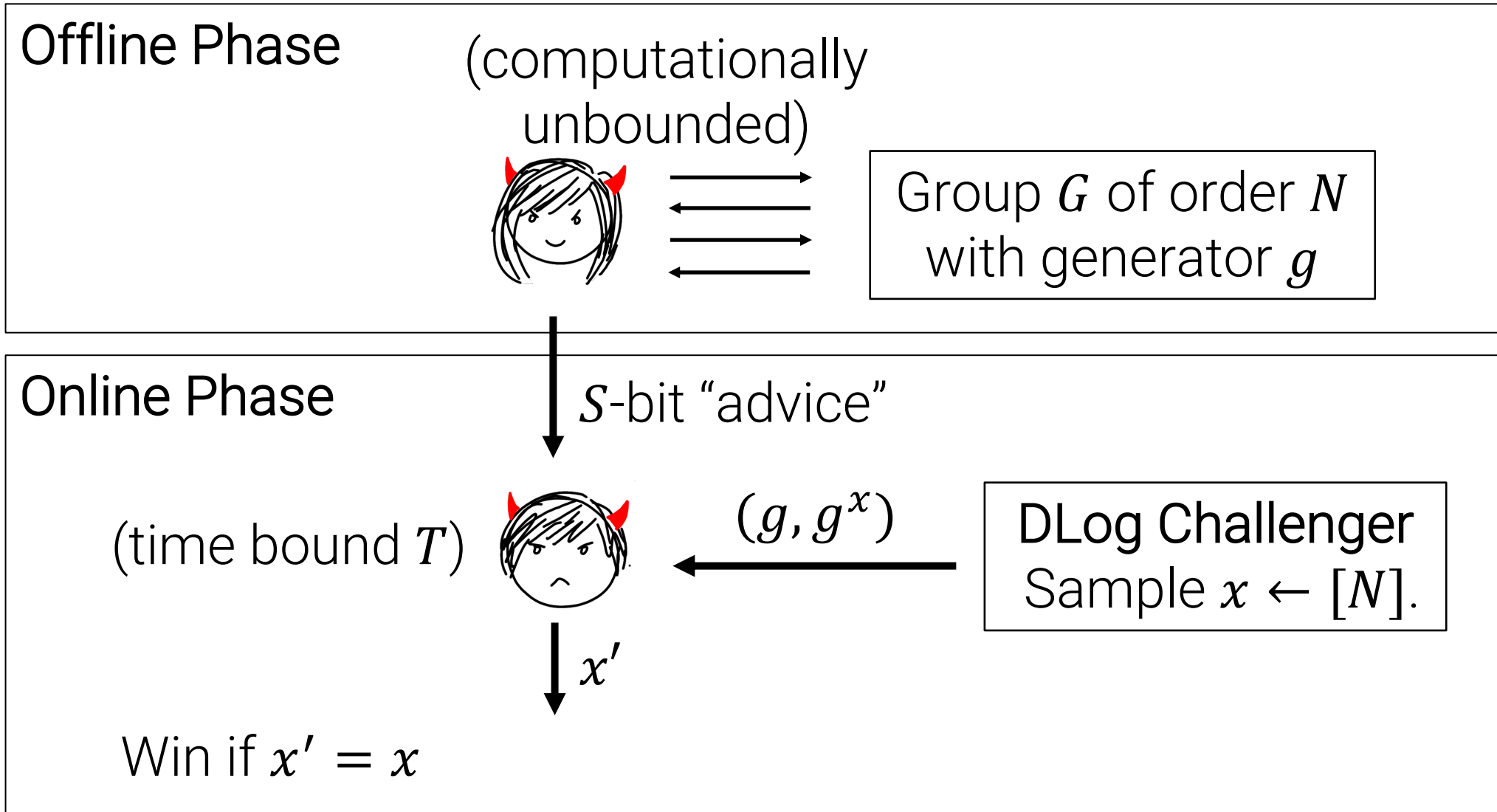
[YYHK14]: These imply

- multiparty non-interactive key agreement with trusted setup [BS02]
- distributed broadcast encryption [BZ14]

Takeaway: It would be surprising if for any “natural” cryptographic group, random-CDH holds but fixed-CDH does not.

The Fixed vs. Random Distinction for Generic Preprocessing Adversaries

Preprocessing Attacks on fixed-DLog



Preprocessing Attacks on fixed-DLog

[Mih10, LCH11, BL13, CK18]: In groups of order N with S bits of advice, online time T , can solve fixed-DLog with probability

$$\epsilon = \Omega\left(\frac{ST^2}{N}\right).$$

Preprocessing Attacks on fixed-DLog

[Mih10, LCH11, BL13, CK18]: In groups of order N with S bits of advice, online time T , can solve fixed-DLog with probability

$$\epsilon = \Omega\left(\frac{ST^2}{N}\right).$$

[CK18]: A generic adversary succeeds with probability at most

$$\epsilon = \tilde{O}\left(\frac{ST^2}{N}\right).$$

Preprocessing Attacks on fixed-DLog

[Mih10, LCH11, BL13, CK18]: In groups of order N with S bits of advice, online time T , can solve fixed-DLog with probability

$$\epsilon = \Omega\left(\frac{ST^2}{N}\right).$$

[CK18]: A generic adversary succeeds with probability at most

$$\epsilon = \tilde{O}\left(\frac{ST^2}{N}\right).$$

Observation: [CK18] is only tight for fixed-DLog.

Claim: Preprocessing algorithms have a lower success probability in the **random**-DLog setting.

To solve **random**-DLog, either 1) ignore preprocessing advice or 2) use preprocessing advice to solve **two fixed**-Dlog instances:

$$\epsilon = \Omega\left(\frac{T^2}{N} + \left(\frac{ST^2}{N}\right)^2\right).$$

success of baby-step-giant-step algorithm

success probability for two fixed-DLog instances

This work: A generic adversary solves **random**-DLog with probability at most

$$\epsilon = \tilde{O} \left(\frac{T^2}{N} + \left(\frac{ST^2}{N} \right)^2 \right).$$

To solve **random**-DLog, either 1) ignore preprocessing advice or 2) use preprocessing advice to solve **two fixed**-Dlog instances:

$$\epsilon = \Omega \left(\frac{T^2}{N} + \left(\frac{ST^2}{N} \right)^2 \right).$$

success of baby-step-giant-step algorithm

success probability for two fixed-DLog instances

This work: A generic adversary solves **random**-DLog with probability at most

$$\epsilon = \tilde{O} \left(\frac{T^2}{N} + \left(\frac{ST^2}{N} \right)^2 \right).$$

Also in the paper: Tight bounds for CDH.

Takeaway: Everything else equal, pre-processing attacks succeed with lower probability on random-generator variants of DLog/CDH.

The Fixed vs. Random Distinction in Assumptions over Non-Uniform Exponents

Assumptions over Non-Uniform Exponents

DDH-II [Canetti97]

If x is drawn from *any well-spread* distribution*,

$$(g, g^x, g^y, g^{xy}) \approx_c (g, g^x, g^y, g^z)$$

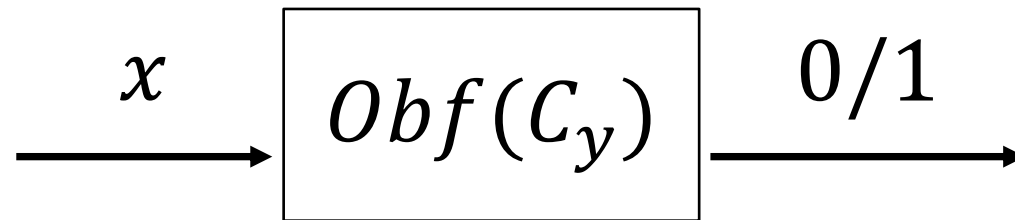
for uniformly random $y, z \leftarrow \mathbb{Z}_q$.

**super-logarithmic min-entropy (hard to guess)*

[Canetti97] shows DDH-II implies obfuscation for point functions.

Point Function Obfuscation

$$C_y(x) = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{if } x \neq y \end{cases}$$

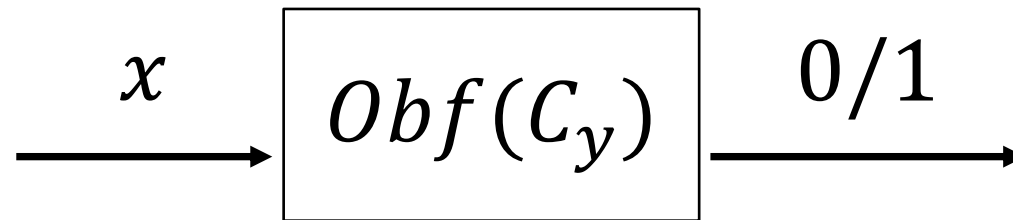


Security: Implementation of $Obf(C_y)$ should hide y

Point Function Obfuscation

[Wee05] proves that strong assumptions are necessary for point function obfuscation

$$C_y(x) = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{if } x \neq y \end{cases}$$



Security: Implementation of $Obf(C_y)$ should hide y

Non-Malleable Point Function Obfuscation [CV08]

[KY18] **Observation:** Given a [Canetti97] obfuscation $Obf(C_y)$, adversary can “maul” to get obfuscation that accepts on related point $f(y)$, i.e. $Obf(C_{f(y)})$.

[KY18] **Goal:** Make $Obf(C_y)$ non-malleable.

Strong Power DDH [KY18]

If x is drawn from *any well-spread* distribution*,

$$(g, g^x, g^{x^2}, \dots, g^{x^k}) \approx_c (g, g^{r_1}, g^{r_2}, \dots, g^{r_k})$$

for uniformly random $r_1, r_2, \dots, r_k \leftarrow \mathbb{Z}_q$.

**super-logarithmic min-entropy (hard to guess)*

[KY18] shows Strong Power DDH implies non-malleable obfuscation for point functions.

This work: Revisiting Non-Malleable Point Obfuscation

Strong Power DDH [KY18]

If x is drawn from *any well-spread* distribution*,

$$(g, g^x, g^{x^2}, \dots, g^{x^k}) \approx_c (g, g^{r_1}, g^{r_2}, \dots, g^{r_k})$$

for uniformly random $r_1, r_2, \dots, r_k \leftarrow \mathbb{Z}_q$.

**super-logarithmic min-entropy (hard to guess)*

Observation: If g is a fixed generator, the assumption is false

Pick x so that g^x begins with 0.

Non-Malleable Point Obfuscation from a New Assumption

Our New Assumption (a toy version)

If x is drawn from any well-spread distribution and $a, r \leftarrow \mathbb{Z}_q$

$$(a, g^{ax+x^2}) \approx_c (a, g^r).$$

Non-Malleable Point Obfuscation from a New Assumption

Our New Assumption (a toy version)

If x is drawn from any well-spread distribution and $a, r \leftarrow \mathbb{Z}_q$

$$(a, g^{ax+x^2}) \approx_c (a, g^r).$$

Theorem: Our assumption holds in the generic group model, even if the distribution is picked after the generic group labels are fixed.

Non-Uniform Assumptions in the Generic Group Model

- All existing generic group proofs of DDH-II assume the generic group labeling function is sampled *independently* of the well-spread distribution.
- This enables proving false assumptions hold in the GGM!

Non-Uniform Assumptions in the Generic Group Model

- All existing generic group proofs of DDH-II assume the generic group labeling function is sampled *independently* of the well-spread distribution.
- This enables proving false assumptions hold in the GGM!
- We give a new GGM proof of DDH-II where the well-spread distribution is picked *after* the labeling is fixed.

Thank you!

Questions?

slides: cs.princeton.edu/~fermim/talks/crypto-2019.pdf

character art: Eysa Lee