An Analysis of the Algebraic Group Model

Jonathan Katz
University of Maryland

Cong Zhang
Zhejiang University

Hong-Sheng Zhou
Virginia Commonwealth University

https://eprint.iacr.org/2022/210

Outline

- Background
  - Generic Group Model (GGM)
  - Algebraic Group Model (AGM)
- Our result: Analysis of the AGM
  - Issue #1
  - Issue #2
- Thoughts

Background: (Cyclic) Group based Crypto

- Diffie-Hellman 1976
- Security of a crypto scheme/protocol,
  can be based on an appropriate ***hardness assumption*** *relative to a group*
- Encodings matter

Background: Group Encodings

- Group encodings
  - Consider encoding $\sigma : \mathbb{Z}_p \to \{0,1\}^\ell$ $\ell \geq \lceil \log p \rceil$
  - id = trivial encoding, i.e, a binary integer; addition mod p
- Group encodings matter
  - DLOG hard: secure prime $q = 2p + 1$

    order-p subgroup of $\mathbb{Z}_q^*$

    multiplication modulo q
  - DLOG trivial: $\mathbb{Z}_p$

    addition modulo p

Background: Security Games

- Code-based security games (Bellare-Rogaway, Eurocrypt 2006)

- Game $\mathbf{G}_\sigma$, parameterized by encoding σ, played by algorithm A

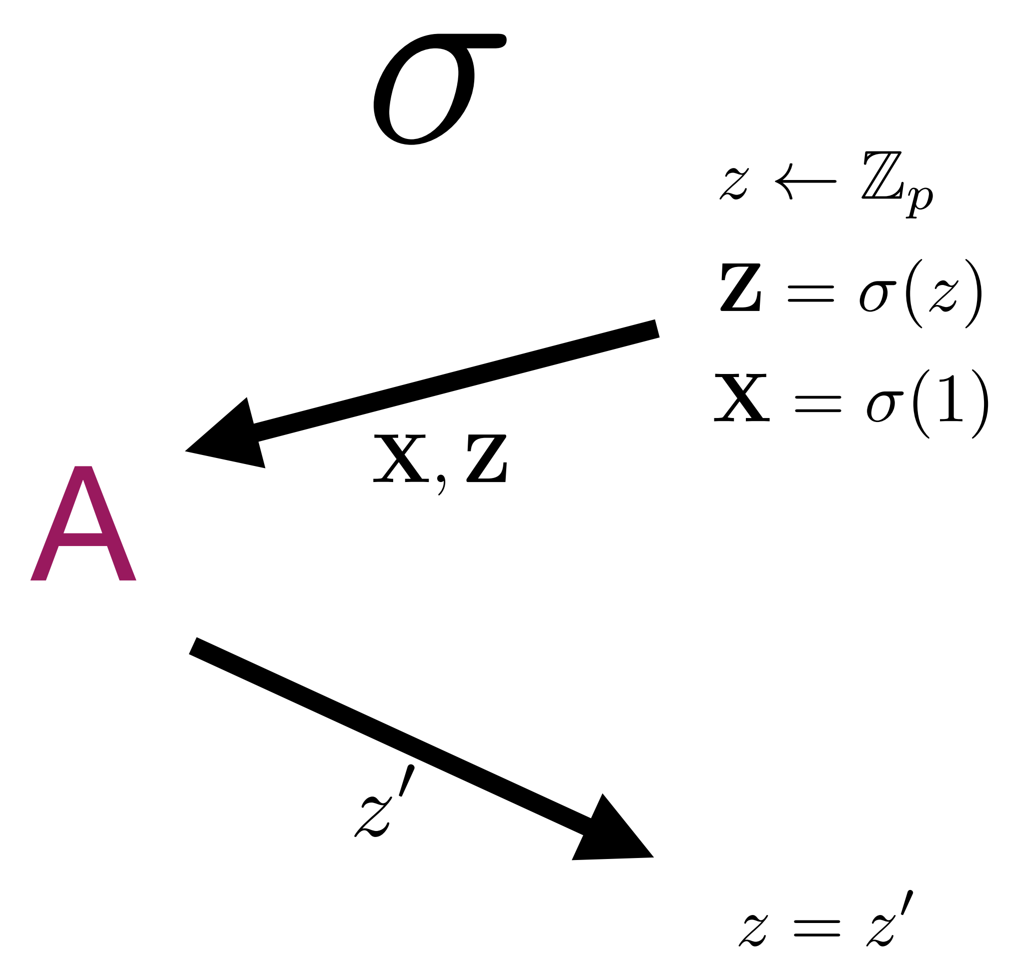
- Algorithm A succeeds if $\mathbf{G}_\sigma^{\mathsf{A}} = 1$

$$\mathbf{Succ}_{\mathbf{G}_\sigma}^{\mathsf{A}} \stackrel{\text{def}}{=} \Pr[\mathbf{G}_\sigma^{\mathsf{A}} = 1]$$

$\mathbf{dlog}_\sigma^{\mathsf{A}}$

01 $z \leftarrow \mathbb{Z}_p$
02 $z' \leftarrow \mathsf{A}(\sigma(1), \sigma(z))$
03 Return 1 iff $z' = z$

The discrete-logarithm game **dlog**



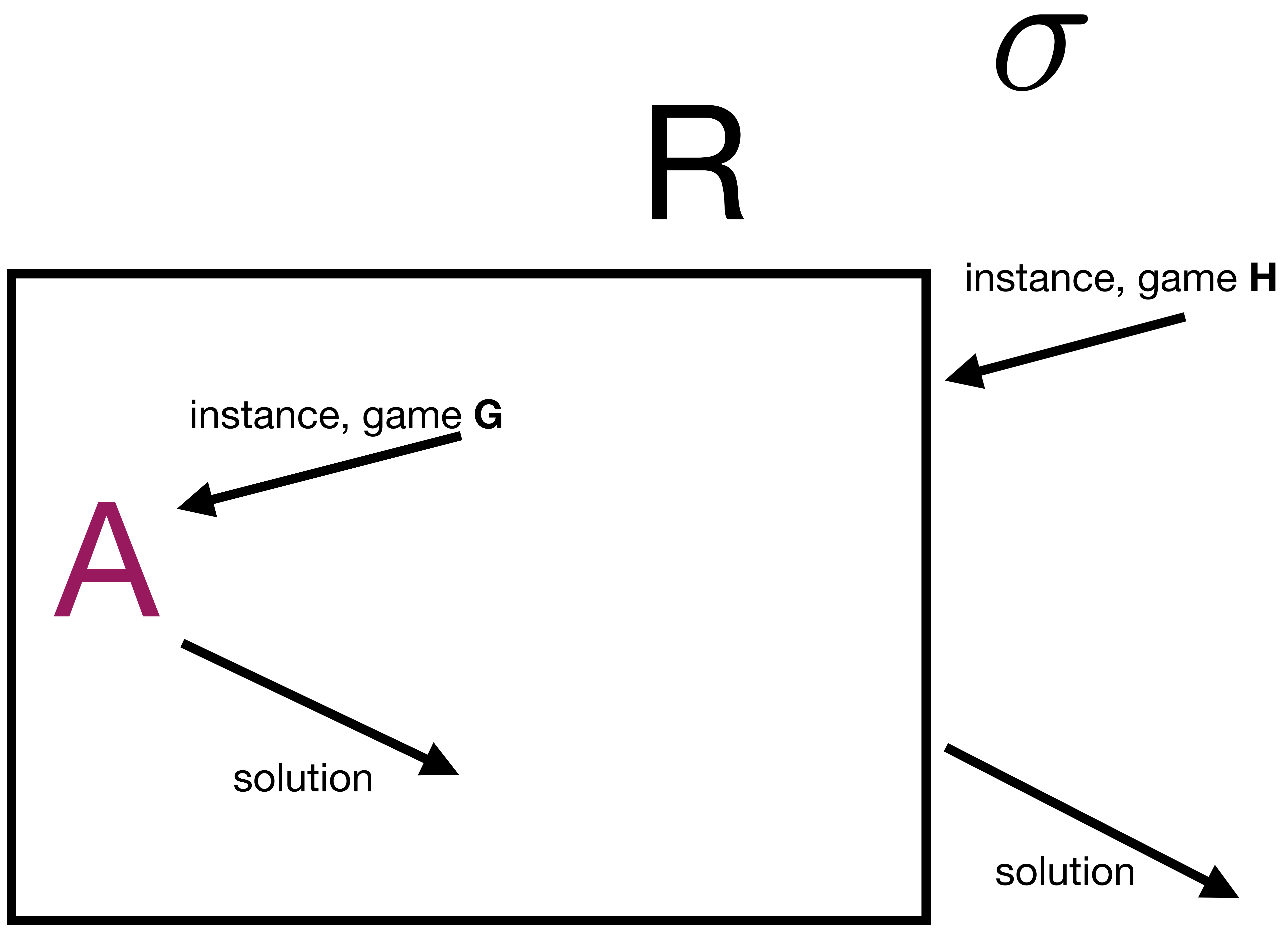
Background: Security Reductions

Let $\mathbf{G}_\sigma, \mathbf{H}_\sigma$ be security games

$$\mathsf{B} := \mathsf{R}^{\mathsf{A}}$$

$$\mathbf{H}_\sigma \xRightarrow{(\Delta_t, \Delta_\epsilon)} \mathbf{G}_\sigma$$

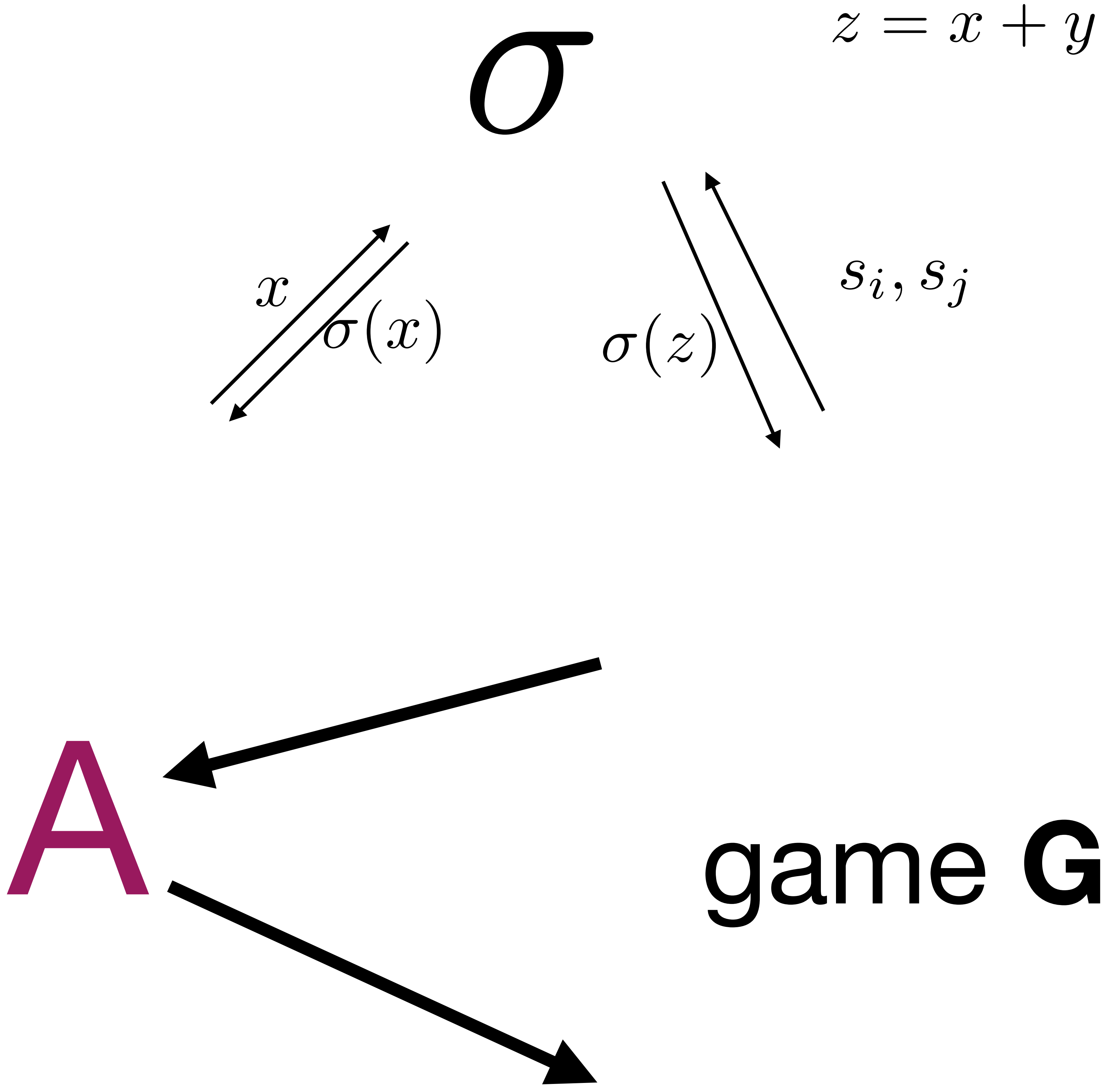
$$\mathbf{Succ}^{\mathsf{B}}_{\mathbf{H}_\sigma} \geq \frac{1}{\Delta_\epsilon} \cdot \mathbf{Succ}^{\mathsf{A}}_{\mathbf{G}_\sigma}, \quad \mathbf{Time}^{\mathsf{B}}_{\mathbf{H}_\sigma} \leq \Delta_t \cdot \mathbf{Time}^{\mathsf{A}}_{\mathbf{G}_\sigma}$$



Background: Generic Group Model

Shoup 1997

$z = x + y \bmod p \quad \text{if } s_i = \sigma(x) \wedge s_j = \sigma(y)$



Algebraic Group Model

- Algebraic Group Model (AGM)
  - Fuchsbauer-Kiltz-Loss 2018 [FKL18]
- any group elements output by an algorithm must be accompanied by a *representation* relative to the ordered set *S* of group elements (the base set) provided to that algorithm as input

base set

$S = (\sigma(x_1), \ldots, \sigma(x_k))$

A

game **G**

group element $\sigma(y) = \prod_i \sigma(x_i)^{r_i}$

representation $\vec{r} = (r_1, \ldots, r_k) \in \mathbb{Z}_p^k$

Algebraic Group Model

- Algebraic Group Model (AGM)
  - Fuchsbauer-Kiltz-Loss 2018

Lemma. *Let* $\mathbf{G}$ *and* $\mathbf{H}$ *be algebraic security games such that*

- $\mathbf{H} \xrightarrow{(\Delta_t, \Delta_\epsilon)}_{\mathsf{alg}} \mathbf{G}$;
- $\mathbf{H}$ *is* (t, ϵ)*-hard in the GGM;*

Then $\mathbf{G}$ *is* $(t/\Delta_t, \epsilon \cdot \Delta_\epsilon)$*-hard in the GGM.*

Outline

- Background
  - Generic Group Model (GGM)
  - Algebraic Group Model (AGM)
- Our result: Analysis of the AGM
  - **Issue #1**
  - Issue #2
- Thoughts

Analysis of the AGM

- Definition and intuition, mismatched:
  - Intuition in [FKL18]
    "the only way for an algebraic algorithm to output a new group element is to derive it via group multiplication from known group elements"
  - new group element using non-group operations, along with a valid representation

$\underline{\mathsf{A}(1)}$
$r_1, r_2 \leftarrow \mathbb{Z}_p$
$s \leftarrow r_1 \cdot r_2 \bmod p$
Output (s, s)

Algorithm A wrt the identity encoding id

Outline

- Background
  - Generic Group Model (GGM)
  - Algebraic Group Model (AGM)
- Our result: Analysis of the AGM
  - Issue #1
  - **Issue #2**
- Thoughts

Issue #2

- Algebraic Group Model (AGM)
  - Fuchsbauer-Kiltz-Loss 2018

Lemma. *Let* $\mathbf{G}$ *and* $\mathbf{H}$ *be algebraic security games such that*

- $\mathbf{H} \xrightarrow{(\Delta_t, \Delta_\epsilon)}_{\mathsf{alg}} \mathbf{G}$;
- $\mathbf{H}$ *is* (t, ϵ)*-hard in the GGM;*

Then $\mathbf{G}$ *is* $(t/\Delta_t, \epsilon \cdot \Delta_\epsilon)$*-hard in the GGM.*

Issue #2

- Algebraic Group Model (AGM)
  - Fuchsbauer-Kiltz-Loss 2018

Lemma. *Let* $\mathbf{G}$ *and* $\mathbf{H}$ *be algebraic security games such that*

- $\mathbf{H} \xRightarrow{(\Delta_t, \Delta_\epsilon)}_{\mathsf{alg}} \mathbf{G}$;
- $\mathbf{H}$ *is* (t, ϵ)*-hard in the GGM;*

Then $\mathbf{G}$ *is* $(t/\Delta_t, \epsilon \cdot \Delta_\epsilon)$*-hard in the GGM.*

- We show: A counterexample

$\mathbf{beg}_\sigma^{\mathsf{A}}$

$z \leftarrow \mathbb{Z}_p$
parse $\mathbf{Z} = \sigma(z)$ as the bitstring $z_1 \cdots z_\ell$
$(\mathbf{X}, \mathbf{U}_1, \ldots, \mathbf{U}_\ell) := (\sigma(1), \sigma(z_1), \ldots, \sigma(z_\ell))$
$\mathbf{Z}' \leftarrow \mathsf{A}(\mathbf{X}, \mathbf{U}_1, \ldots, \mathbf{U}_\ell)$
Return 1 iff $(\mathbf{Z}' = \mathbf{Z})$

beg = **b**inary **e**ncoding **g**ame

Theorem. *There are security games* $\mathbf{G}$ *and* $\mathbf{H}$ *such that*

- $\mathbf{H} \xRightarrow{(2,1)}_{\mathsf{alg}} \mathbf{G}$;
- $\mathbf{H}$ *is* $(t, O(t^2/p))$*-hard with respect to Shoup-generic algorithms;*
- *There is a Shoup-generic algorithm* A *running in time* $O(\ell)$ *with* $\mathbf{Succ}_{\mathbf{G}}^{\mathsf{A}} = 1$.

Issue #2

- We show: A counter example

G=

$\mathbf{beg}_\sigma^{\mathsf{A}}$

01 $z \leftarrow \mathbb{Z}_p$
02 parse $\mathbf{Z} = \sigma(z)$ as the bitstring $z_1 \cdots z_\ell$
03 $(\mathbf{X}, \mathbf{U}_1, \ldots, \mathbf{U}_\ell) := (\sigma(1), \sigma(z_1), \ldots, \sigma(z_\ell))$
04 $\mathbf{Z}' \leftarrow \mathsf{A}(\mathbf{X}, \mathbf{U}_1, \ldots, \mathbf{U}_\ell)$
05 Return 1 iff $(\mathbf{Z}' = \mathbf{Z})$

H=

$\mathbf{dlog}_\sigma^{\mathsf{A}}$

01 $z \leftarrow \mathbb{Z}_p$
02 $z' \leftarrow \mathsf{A}(\sigma(1), \sigma(z))$
03 Return 1 iff $z' = z$

Theorem. *There are security games* $\mathbf{G}$ *and* $\mathbf{H}$ *such that*

- $\mathbf{H} \overset{(2,1)}{\Longrightarrow}_{\mathsf{alg}} \mathbf{G}$;
- $\mathbf{H}$ *is* $(t, O(t^2/p))$*-hard with respect to Shoup-generic algorithms;*
- *There is a Shoup-generic algorithm* A *running in time* $O(\ell)$ *with* $\mathbf{Succ}_{\mathbf{G}}^{\mathsf{A}} = 1$.

$\mathbf{beg}_\sigma^{\mathsf{A}}$

```
z ← Z_p
parse Z = σ(z) as the bitstring z_1 ··· z_ℓ
(X, U_1, ..., U_ℓ) := (σ(1), σ(z_1), ..., σ(z_ℓ))
Z' ← A(X, U_1, ..., U_ℓ)
Return 1 iff (Z' = Z)
```

σ

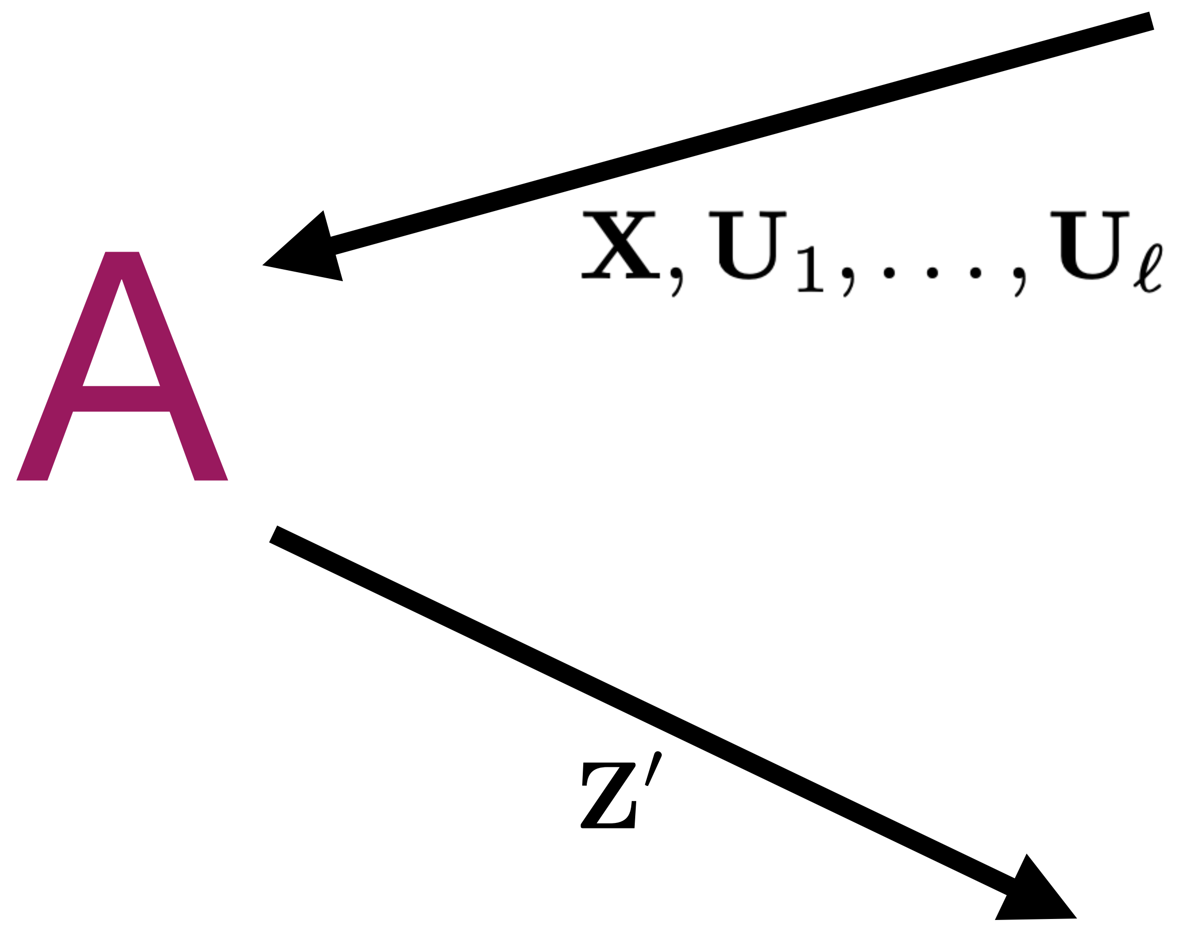
$z \leftarrow \mathbb{Z}_p$

$\mathbf{Z} = \sigma(z)$

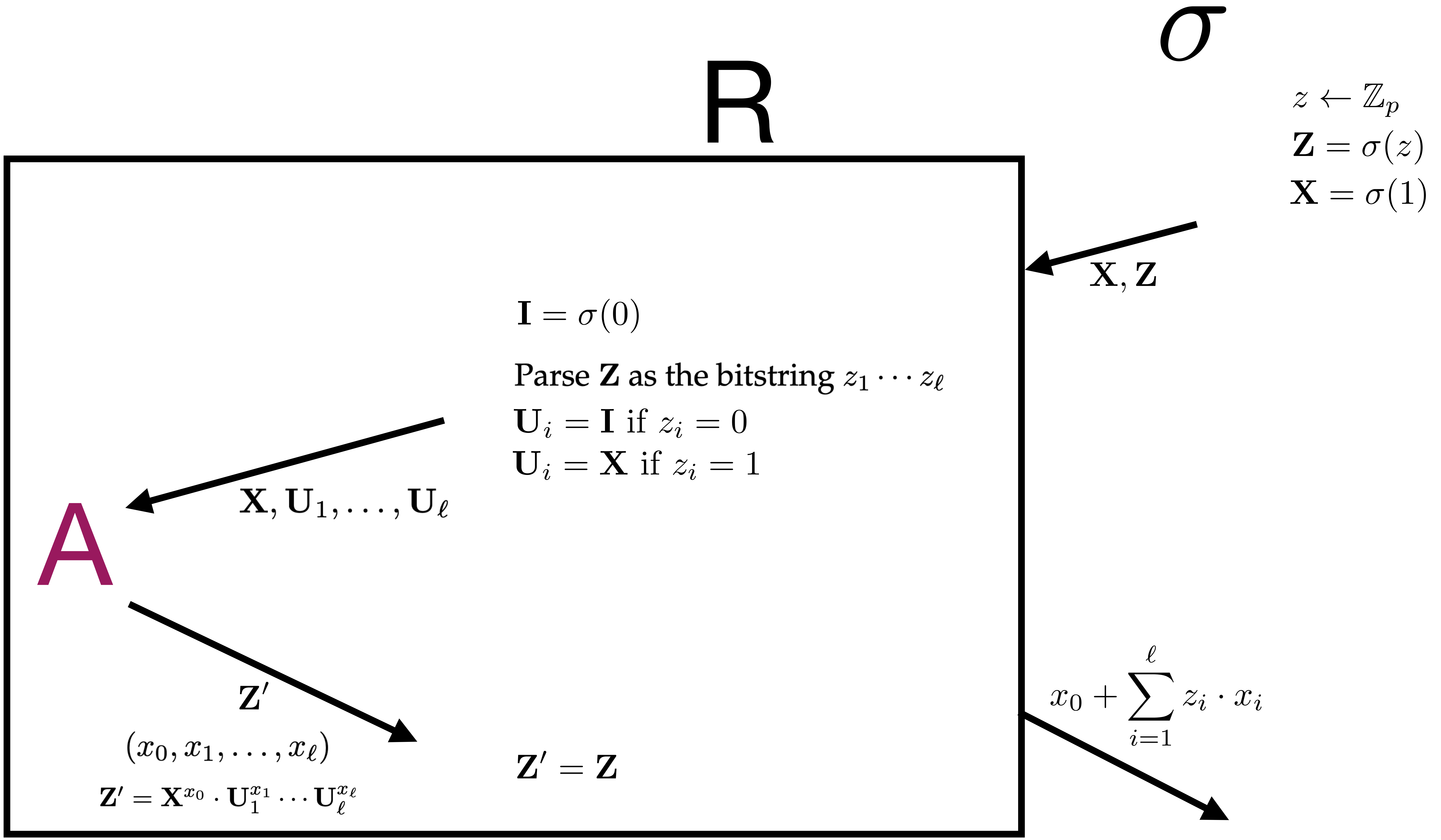
Parse $\mathbf{Z}$ as the bitstring $z_1 \cdots z_\ell$

$(\mathbf{U_1}, \ldots, \mathbf{U}_\ell) = (\sigma(z_1), \ldots, \sigma(z_\ell))$

$\mathbf{X} = \sigma(1)$



$\mathbf{Z}' = \mathbf{Z}$



σ
$z \leftarrow \mathbb{Z}_p$
$\mathbf{Z} = \sigma(z)$
$\mathbf{X} = \sigma(1)$
R
$\mathbf{X}, \mathbf{Z}$
$\mathbf{I} = \sigma(0)$
Parse $\mathbf{Z}$ as the bitstring $z_1 \cdots z_\ell$
$\mathbf{U}_i = \mathbf{I}$ if $z_i = 0$
$\mathbf{U}_i = \mathbf{X}$ if $z_i = 1$
$\mathbf{X}, \mathbf{U}_1, \ldots, \mathbf{U}_\ell$
A
$\mathbf{Z}'$
$(x_0, x_1, \ldots, x_\ell)$
$\mathbf{Z}' = \mathbf{X}^{x_0} \cdot \mathbf{U}_1^{x_1} \cdots \mathbf{U}_\ell^{x_\ell}$
$\mathbf{Z}' = \mathbf{Z}$
$x_0 + \sum_{i=1}^{\ell} z_i \cdot x_i$

Conclusion and Thoughts

- Analysis of the AGM:
  - it is not clear whether the class of algebraic algorithms contains the class of generic algorithms.
- the main justification for studying reductions in the AGM does not hold in certain settings.
- Future direction ?

Questions?

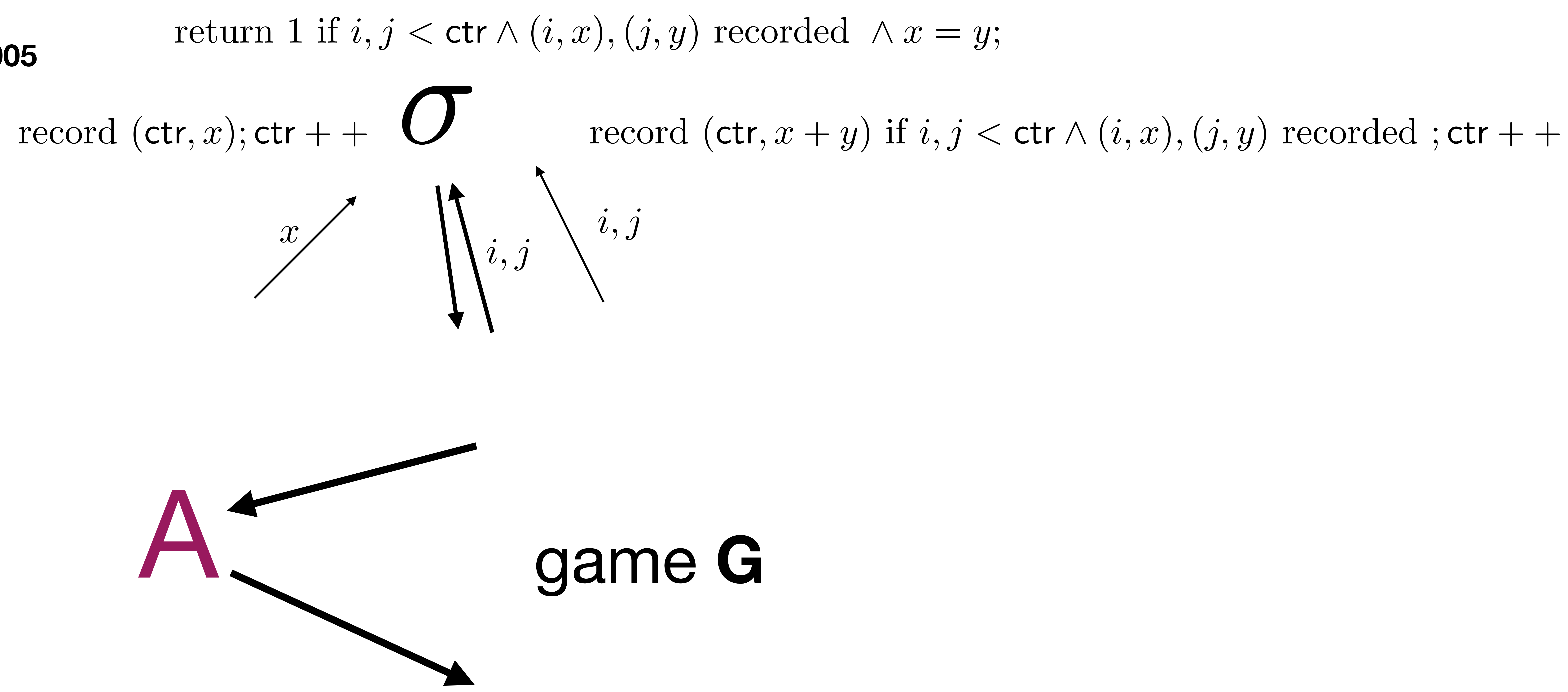
- Thanks for your attention

- https://eprint.iacr.org/2022/210

Backup Slides

Background: Generic Group Model

Maurer 2005



Background: Generic Group Model

Shoup 1997