How to Idealize Generic Groups

Mark Zhandry (NTT Research)

Cryptographic Groups

[Diffie-Hellman'76]

(Cyclic) group $\mathbb{G}$ with efficient multiplication

($g, h \rightarrow g \times h$ easy)

Tons of hardness assumptions:

- Discrete log: $g, g^a \rightarrow a$
- CDH: $g, g^a, g^b \rightarrow g^{ab}$
- DDH: g, g^a, g^b, g^{ab} vs g, g^a, g^b, g^c
- DHI: $g, g^a, g^{a^2}, \cdots g^{a^\ell} \rightarrow g^{1/a}$
- …

Generic/Idealized Groups

For certain well-designed groups, best known practical attacks on many assumptions are *generic* (independent of group itself)

⬇

Generic Group Model (GGM): Only consider adversaries that are independent of group

[Nechaev'94, Shoup'97, Maurer'04]

Shoup'97: Random Labels

Random injection $L : \mathbb{Z}_p \to \{0,1\}^n$

Interpret $L(x)$ as g^x

Adversary computes group operation using oracle:

$$\mathsf{M} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$$

$$\mathsf{M}(L(x), L(y)) = L(x+y)$$

Thm (Informal) [Shoup'97,…]: "Most" interesting problems are hard in Shoup's GGM

Idea: Show that solving problem requires exponentially-many queries to Mult. Query count then lower-bounds running time

Discussion

Many (reasonable) criticisms of generic groups
(e.g. [Fischlin'00, Dent'02, Koblitz-Menezes'06])

Thm [Dent'02, building on Canetti-Goldreich-Halevi'98]: $\exists$ (contrived) assumptions secure in GGM that are insecure in *any* concrete group

Discussion

Due to [Dent’02], generic proofs do not prove actual hardness, but are interpreted as heuristic evidence

Nevertheless, the GGM remains a critical tool in the design of both practical and theoretical constructions. As such, studying GGM is crucial

There is another...

Maurer'05: Pointers/Type Safety

```
Mult(Element h1, Element h2) {
    return new Element(h1.value * h2.value);
}
EqualQ(Element h1, Element h2) {
    return h1.value==h2.value;
}
```

No other operations on `Element` variables allowed

Motivating question for this work:

Which model to use?

Most Literature Treats the Two Equivalently

Generic group model

文A 2 languages

Article Talk Read Edit View history Tools

From Wikipedia, the free encyclopedia

The **generic group model**[1][2] is an idealised cryptographic model, where the adversary is only given access to a randomly chosen encoding of a group, instead of efficient encodings, such as those used by the finite field or elliptic curve groups used in practice.

References [edit]

1. ^ a b Victor Shoup (1997). "Lower bounds for discrete logarithms and related problems" (PDF). *Lecture Notes in Computer Science*. Advances in Cryptology – Eurocrypt '97. Vol. 1233. Springer-Verlag. pp. 256–266. Retrieved 2010-04-09.
2. ^ Ueli Maurer (2005). "Abstract models of computation in cryptography" (PDF). *Lecture Notes in Computer Science*. 10th IMA Conference On Cryptography and Coding. Vol. 2796. Springer-Verlag. pp. 1–12. Archived from the original (PDF) on 2017-07-06. Retrieved 2007-11-01.

Maybe it doesn't matter?

On the Equivalence of Generic Group Models

Tibor Jager and Jörg Schwenk

Horst Görtz Institute for IT Security
Ruhr-University Bochum, Germany

Abstract. The *generic group model* (GGM) is a commonly used tool in cryptography, especially in the analysis of fundamental cryptographic … notion, it is not … that … security proof in one model implies security in the other model. Thus the validity of a proven statement may depend on the choice of the model. In this paper we prove the equivalence of the models proposed by Shoup [2] and Maurer [3].

But...

Does Fiat-Shamir Require a Cryptographic Hash Function?

Yilei Chen* Alex Lombardi† Fermi Ma‡ Willy Quach§

(Shoup's)

First we explain why Fiat-Shamir for Schnorr is secure in the (plain) GGM, even for simple, information-theoretic hash functions. We start with the case of "no-message" signatures (non-interactive identifi-

On the Impossibility of Purely Algebraic Signatures

Nico Döttling[1], Dominik Hartmann[2], Dennis Hofheinz[3], Eike Kiltz[2], Sven Schäge[4], and Bogdan Ursu[3]

in hidden-ord ttings, we have to restric our defi ion o alge aic signatures a lit e). More explicitly, we show:

– the insecurity of all algebraic signature schemes in Maurer's generic group model (in pairing-free groups), as long as these schemes do not rely on other cryptographic assumptions, such as hash functions.

This Talk

eprint 2022/226

1. Comparing Maurer vs Shoup models

2. Comparison to *Algebraic* Group Model (AGM)
   [Fuchsbauer-Kiltz-Loss'18]

3. Generic quantum models for group *actions*

eprint 2023/1097

Part 1: Maurer vs Shoup

TLDR

[Shoup'97] $\geq$ [Maurer'05]

<u>When in doubt, choose Shoup</u>

More nuanced summary

Black-box impossibilities		[Shoup'97]	>>	[Maurer'05]
Security proofs	Single-stage games	[Shoup'97]	=	[Maurer'05]
	Multi-stage games	[Shoup'97]	>>	[Maurer'05]

More nuanced summary

Black-box impossibilities

Typical definitions (e.g. PRGs, PRFs, PKE, Signatures, etc)

aurer'05]

Security proofs

Single-stage games

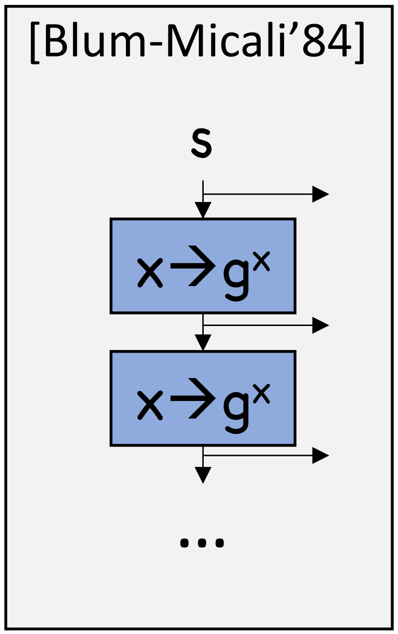
[Shoup'97] = [Maurer'05]

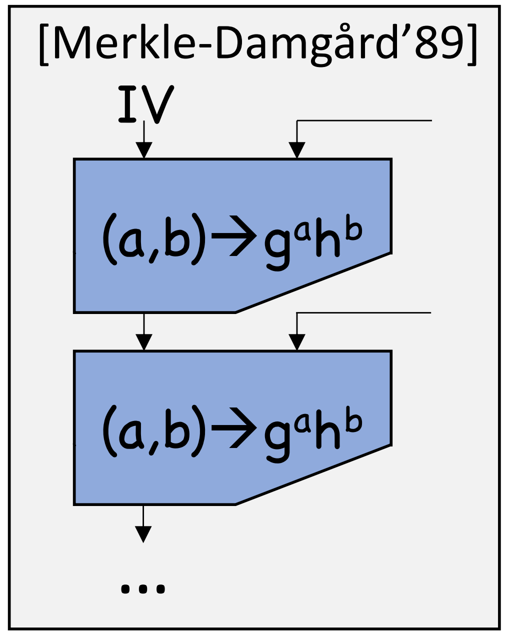
Multi-stage games

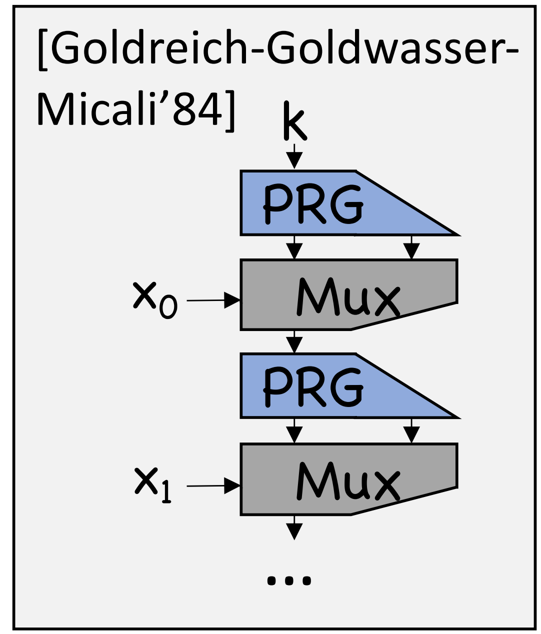
[Shoup'97] ≠ [Maurer'05]

E.g. deterministic encryption, leakage resilience, etc

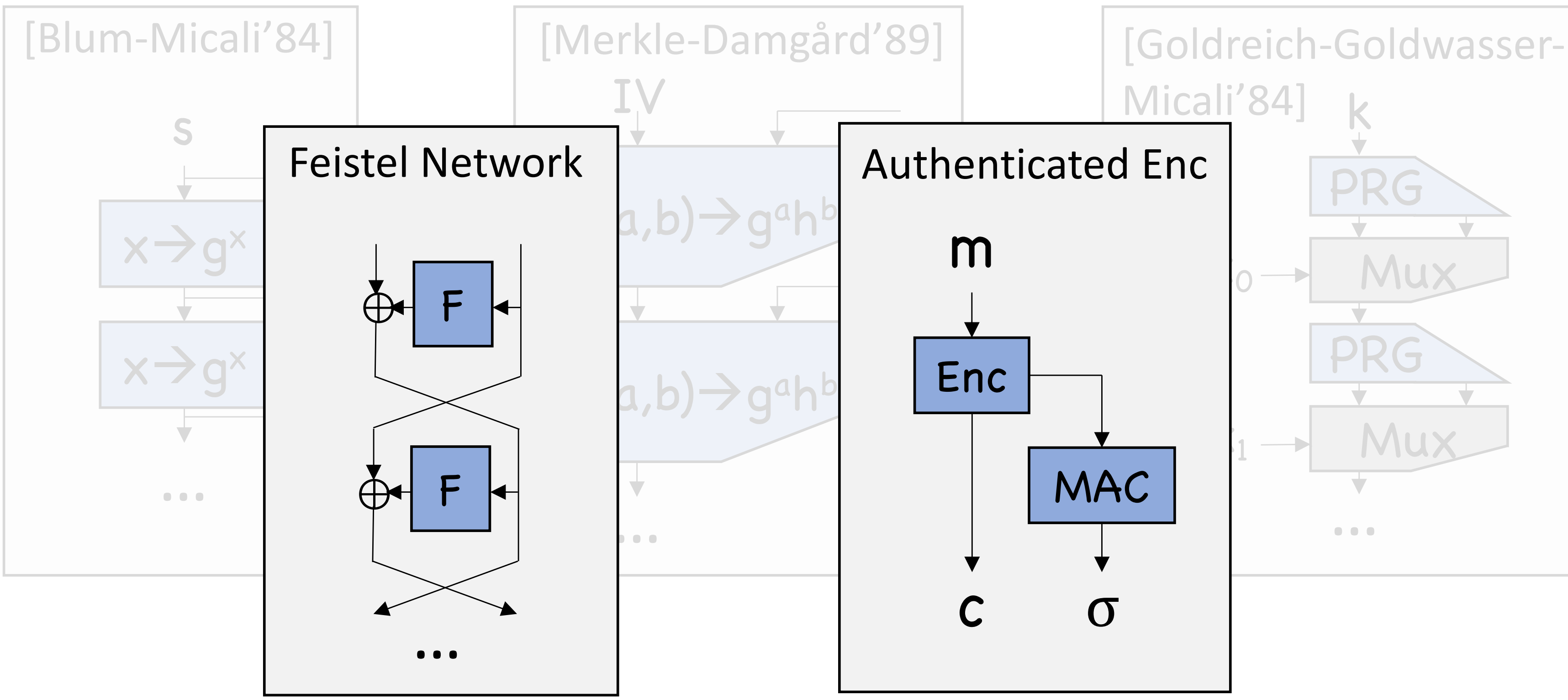
Starting observation: textbook techniques that fail in Maurer



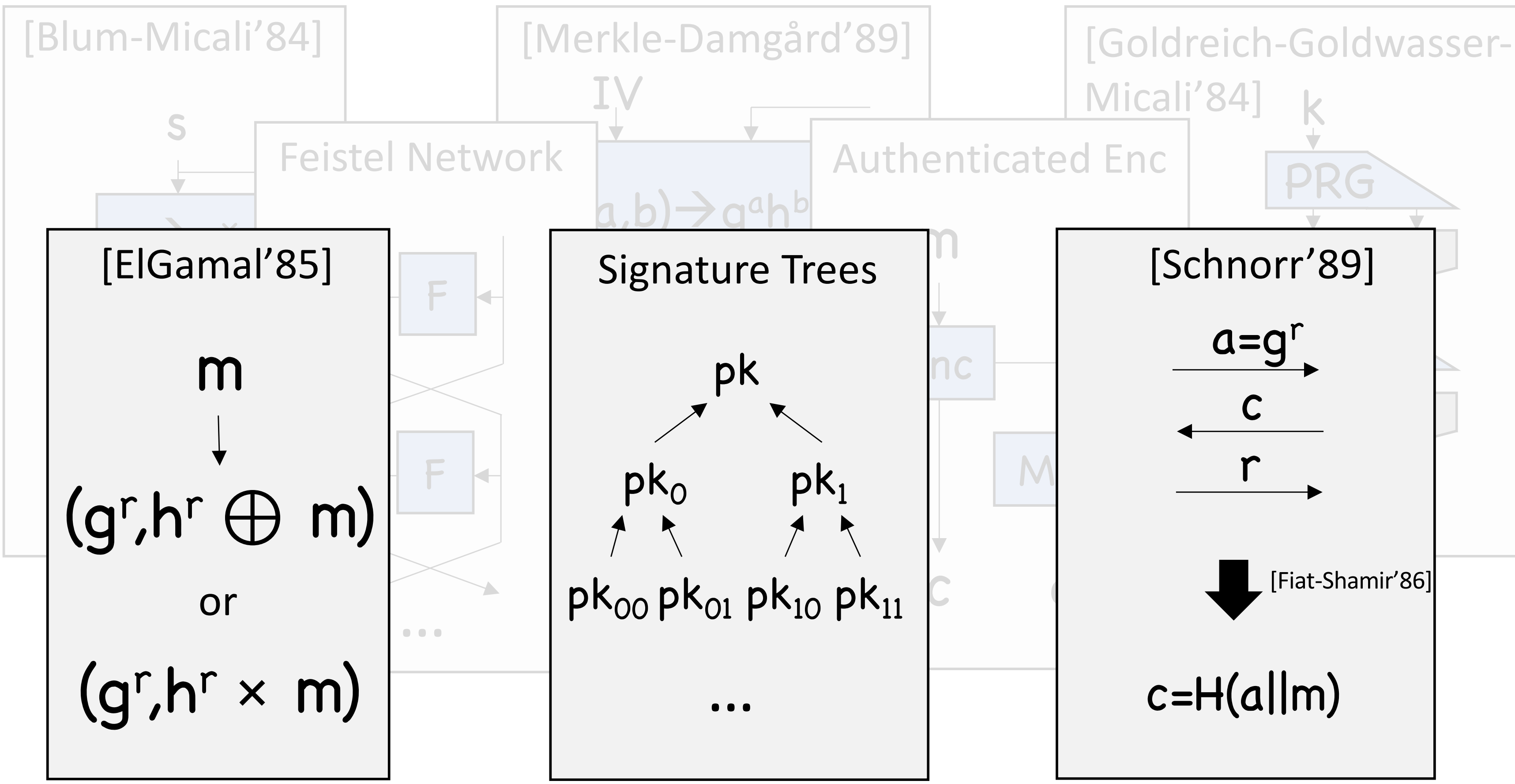




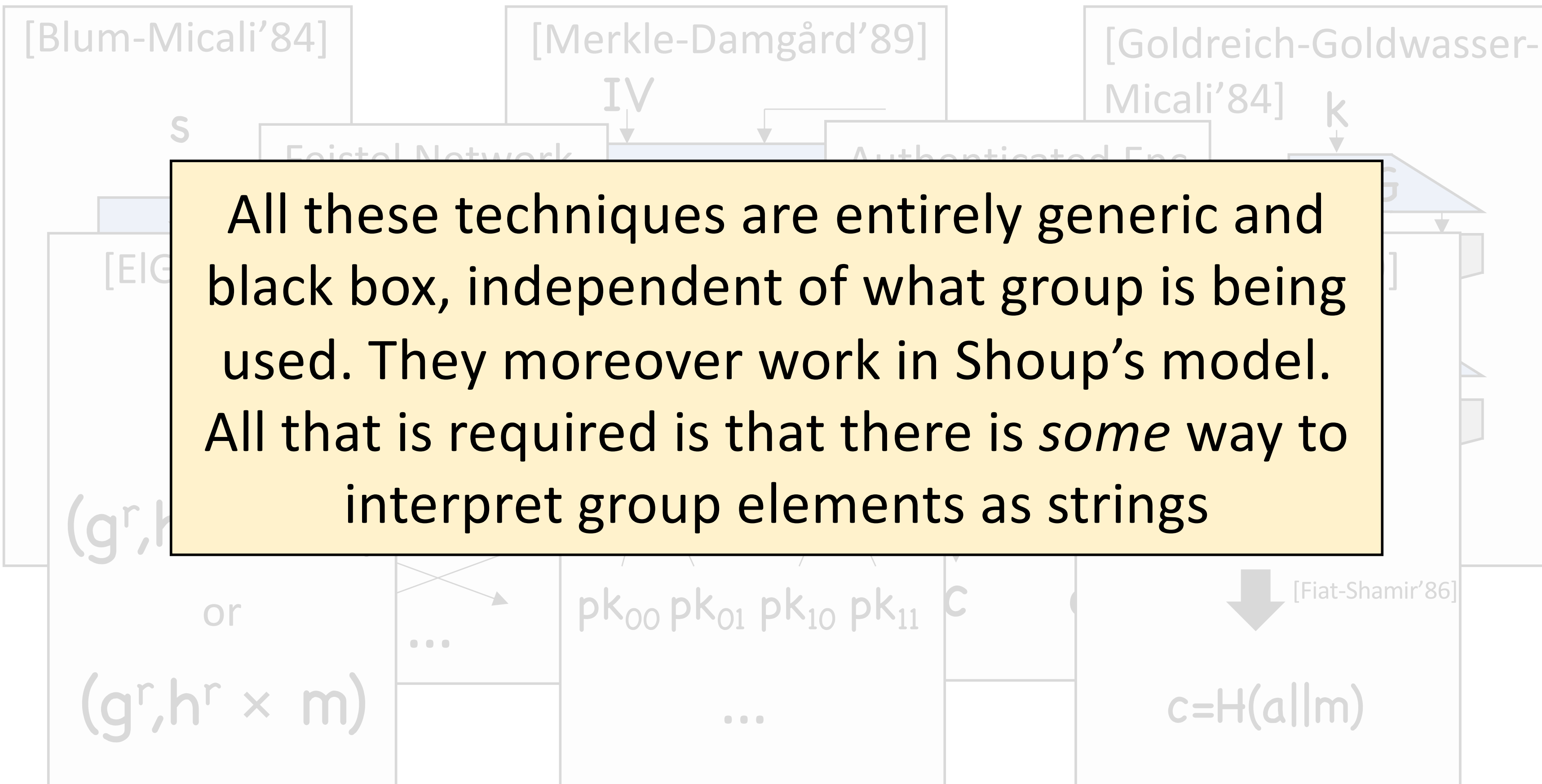
Starting observation: textbook techniques that fail in Maurer



Starting observation: textbook techniques that fail in Maurer



Starting observation: textbook techniques that fail in Maurer



All these techniques are entirely generic and black box, independent of what group is being used. They moreover work in Shoup's model. All that is required is that there is *some* way to interpret group elements as strings

Shoup $\gg$ Maurer for *Impossibilities*

Thm (Implicit from [Chen-Lombardi-Ma-Quach'20]+[Döttling-Hartmann-Hofheinz-Kiltz-Schäge-Ursu'21], formalized and extended in our work): There exist generic and textbook primitives that work in Shoup and standard models, but do not exist in Maurer (e.g. PRPs, unbounded CRHFs, rate-1 encryption)

Thm (our work): Any construction that works in Maurer also works in Shoup

Black box separations in Maurer must be taken with grain of salt

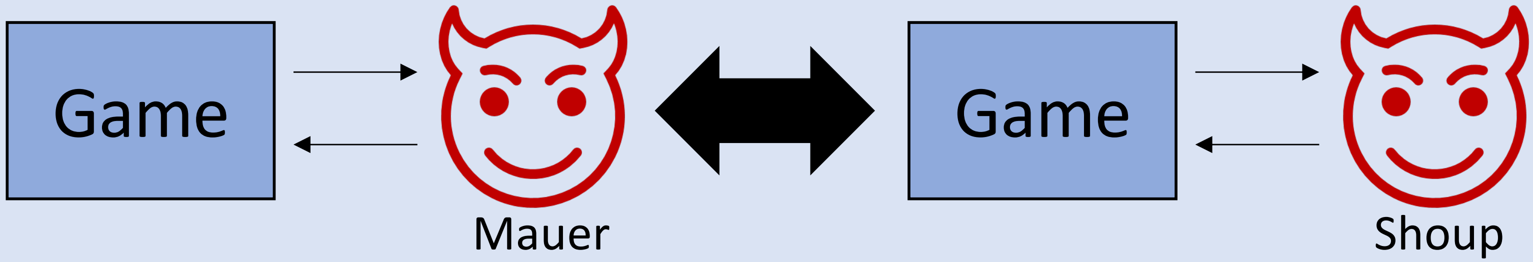
So what’s the deal with Jager-Schwenk?

Historical note: Generic groups originally only used for analyzing hardness of computational problems. Use for *impossibilities* came later

[Dodis-Haitner-Tentes’12, Cramer-Damgård-Kiltz-Zakarias-Zottarel’12, Papakonstantinou-Rackoff-Vahlis’12]

So what’s the deal with Jager-Schwenk?

Thm [Jager-Schwenk’08]:

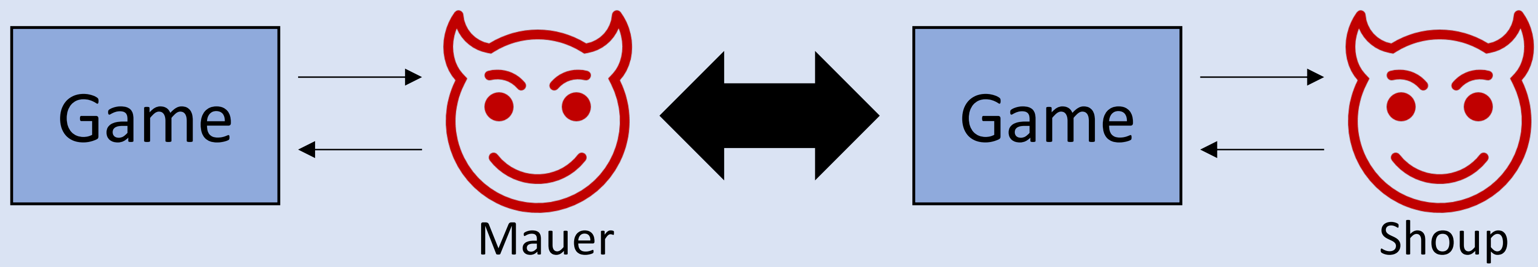


Proof: ➡

```
Mult(Element’ h1, Element’ h2) {
    return new Element’( M(h1.label , h2.label) );
}
EqualQ(Element’ h1, Element’ h2) {
    return h1.label==h2.label;
}
```

So what's the deal with Jager-Schwenk?

Thm [Jager-Schwenk'08]:



Proof: ⬅ lazy sample labelling function

$T =$

E_1	ℓ_1
E_2	ℓ_2
E_3	ℓ_3
E_4	ℓ_4
E_5	ℓ_5

$\mathsf{M}(\ell_x, \ell_y)$:

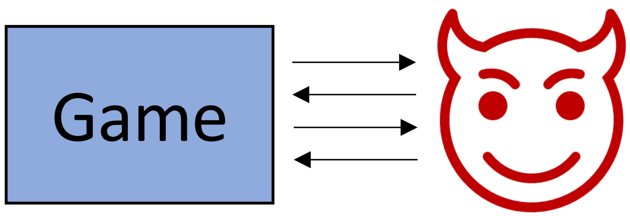
```
Look for (E_x,l_x),(E_y,l_y) in T;
E_z = Mult(E_x, E_y);
Look for (E_z,l_z) in T;
If not found:
        l_z ← {0,1}^n
        Add (E_z, l_z) to T
Output l_z
```

So what's the deal with Jager-Schwenk?

Two Observations:

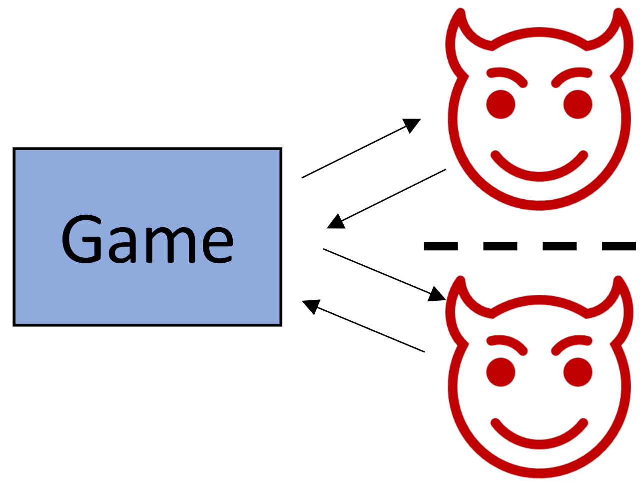
- Jager-Schwenk only makes sense if game makes sense in both models
- Simulation in second case requires keeping state

Single stage



Jager-Schwenk applies

Multi-stage



Jager-Schwenk fails since cannot maintain consistent state between adversaries

Shoup vs Maurer for Proving Security

Thm (our work): Maurer construction → Shoup construction

Thm (our work): For Maurer games, Shoup security → Maurer

Thm (our work): Amongst **single-stage** Maurer games, Maurer security → Shoup security

Re-interpretation of Jager-Schwenk

Thm (our work): ∃ multi-stage Maurer game secure in Maurer but not in Shoup

(Also insecure in any standard-model group)

More nuanced summary

Black-box impossibilities		[Shoup'97]	≫	[Maurer'05]
Security proofs	Single-stage games	[Shoup'97]	=	[Maurer'05]
	Multi-stage games	[Shoup'97]	≫	[Maurer'05]

Part 2: Algebraic Group Model

Algebraic Group Model (AGM) Intuition

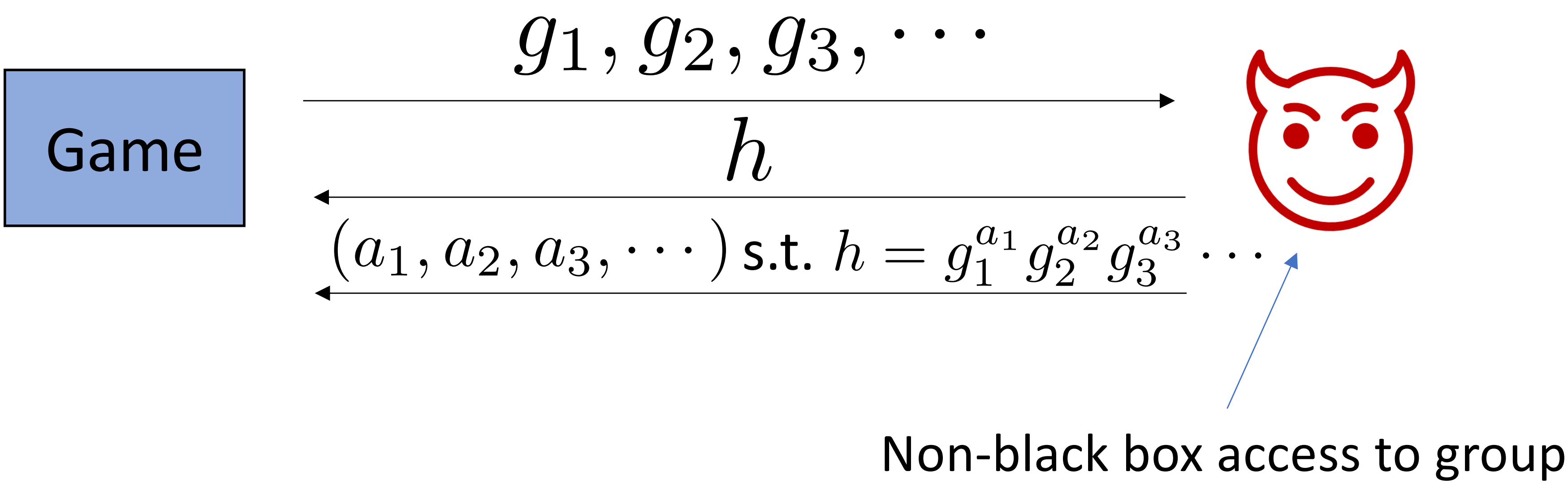
Suppose given $g_1, g_2, g_3, \cdots$

Can construct new group elements as $h = g_1^{a_1} g_2^{a_2} g_3^{a_3} \cdots$ for known $(a_1, a_2, a_3, \cdots)$

For "sufficiently good" groups, seems no other way to generate new group elements

Algebraic Group Model (AGM)

[Fuchsbauer-Kiltz-Loss'18], building on [Paillier-Vergnaud'05]



Often claimed to be "between" generic groups and standard model

Algebraic Group Model (AGM)

[Fuchsbauer-Kiltz-Loss'18], building on [Paillier-Vergnaud'05]

No unconditional security:

AGM does *not* imply that Dlog is hard

(Dlog game doesn't ask for group elements)

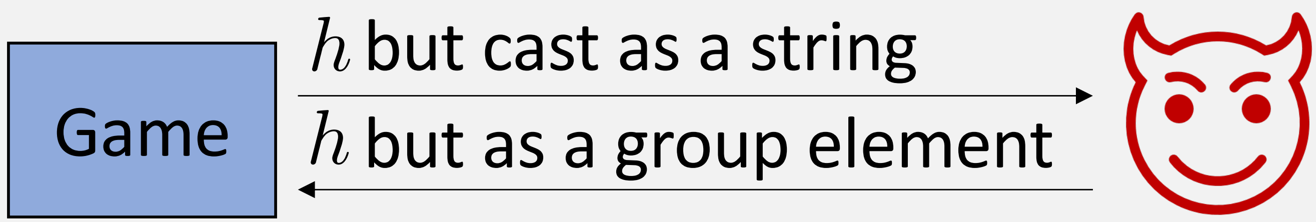
Instead, AGM facilitates *reductions* to assumptions

(e.g. Dlog implies CDH in AGM)

How does AGM compare to GGM?

Observation: AGM not fully defined by FKL

Trivial uninstantiability [FKL]:



Finding representation impossible!

[FKL]: Syntactically distinguish group elements from non-group elements, non-group elements must not “depend” on group elements

What does “depend” mean?

Our position:

AGM only applies to Maurer games

[Katz-Zhang-Zhou'22]:
Different interpretation

Our AGM Results

Consequence of our interpretation and our results:

- AGM is no "worse" than Mauer (and therefore no worse than Shoup for proving single-stage games)
- AGM probably shouldn't be used for black-box impossibilities (not that anyone has advocated for it)

On the other hand, not clear if AGM is actually "better":

Thm (our work): ∃ single-stage Maurer game secure in AGM but not in real world

Open Question

Maurer games that are insecure in GGM

⬇

AGM = GGM

Maurer games that don't ask for group elements

⬇

AGM = standard model

Maurer games secure in AGM under "standard" assumptions

⬇

AGM = GGM

Our take: justifying that AGM > GGM would require finding a game non-trivially outside of these categories

Part 3: Quantum

Quantum Computers Break Groups

[Shor'94]

Suppose $h = g^a$, want to find a

Define $F(x, y) = g^x h^y$

F is periodic: $F((x, y) + (-a, 1)) = F(x, y)$

Thm [Shor'94]: Quantum algorithms can easily find periods

Cryptographic Group *Actions*

[Brassard-Yung'91]

(Abelian) group $\mathbb{G}$ efficiently acting on set $\mathcal{X}$

$$g * (h * x) = (gh) * x$$

Discrete log: $(x, a * x) \rightarrow a$

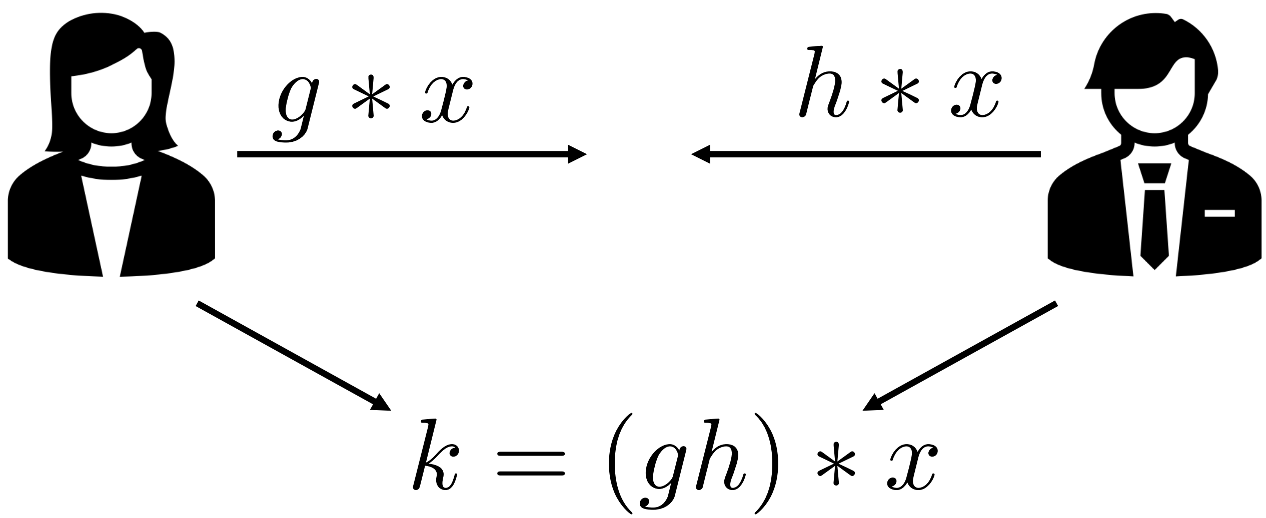
Groups are special case of group actions:

$\mathbb{Z}_p^*$ acts on $\mathbb{G}$ via $a * x = x^a$

Cryptographic Group *Actions*

[Brassard-Yung'91]

Good enough for some cryptosystems, but not others

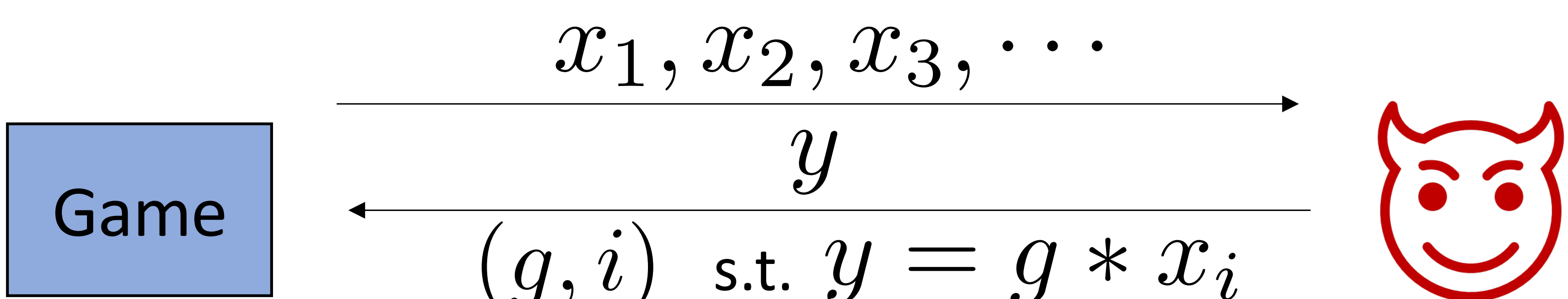


Idealized models for group actions

Not hard to define Shoup, Maurer, AG(A)M models for group actions

[Montgomery-**Z**'22,Liu-Montomgery-**Z**'23, Boneh-Guan-**Z**'23, Duman-Hartmann-Kiltz-Kunzweiler-Lehmann-Riepel'23, Orsini-Zanotto'23]

(Classical) AGAM for group actions:



Using idealized group actions to prove security?

Thm [Ettinger-Høyer'00]: Inefficient but query-bounded quantum algorithm for DLog

(works in Shoup or Maurer)

Don't know how to prove generic lower-bounds except through query complexity

⬇

GGAM for group actions (Shoup or Maurer) useless?

What About Quantum AGAM?

Observation [Duman-Hartmann-Kiltz-Kunzweiler-Lehmann-Riepel'23]: Still meaningful to assume Dlog and use AGAM for reductions, thus advocate for using AGAM for group action security proofs

However…

Problem with quantum AGAM

Recall implicit assumption in (classical) AGM:

If at some point you "knew" some data (e.g. $a_1, a_2, \cdots$), you will always know it

$$g_1, g_2, g_3, \cdots \longrightarrow h = g_1^{a_1} g_2^{a_2} g_3^{a_3} \cdots$$

can also output $a_1, a_2, \cdots$

Problem with quantum AGAM

Analog for quantum data is simply false!

Thm (our work): Can construct quantum superposition over set elements with provably unknown DLogs

In particular, can construct:

$$\frac{1}{\sqrt{|\mathbb{G}|}}\sum_g e^{i2\pi g^2}|g*x\rangle$$

Very different from:

$$\frac{1}{\sqrt{|\mathbb{G}|}}\sum_g e^{i2\pi g^2}|g*x\rangle|g\rangle$$

Using idealized group actions to prove quantum security

Summary:
- Quantumly, AGAM actually incomparable with GGAM
- Should be skeptical of AGAM
- Can't get unconditional hardness in GGAM

Largely open*: maybe GGAM can help prove security based on computational assumption

e.g. Dlog → DDH?

* We give some examples in paper

Thanks!