RUB

RUHR-UNIVERSITÄT BOCHUM

The Algebraic Group Model and its Applications

Crypto, 22.8.18

Georg Fuchsbauer[1], Eike Kiltz[2], and **Julian Loss**[2]

[1] Inria, ENS, CNRS, PSL, France

[2] Ruhr University Bochum, Germany

Horst Görtz Institut für IT-Sicherheit

Idealized Models In Cryptography

- Given cryptographic scheme, in what model do we prove it secure?

Idealized Models In Cryptography

- Given cryptographic scheme, in what model do we prove it secure?
- Best case: proof in *standard model* (no simplifying assumptions).

Idealized Models In Cryptography

- Given cryptographic scheme, in what model do we prove it secure?
- Best case: proof in *standard model* (no simplifying assumptions).
- Often, this is not possible (or just very hard to do).

Idealized Models In Cryptography

- Given cryptographic scheme, in what model do we prove it secure?
- Best case: proof in *standard model* (no simplifying assumptions).
- Often, this is not possible (or just very hard to do).
- Must resort to *idealized models* instead.

Idealizing the Real World– But How?

- Goal: 'abstract away' as many non-essential properties as possible.

Idealizing the Real World– But How?

- Goal: ‘abstract away’ as many non-essential properties as possible.
- Prove statements in simplified, idealized model.

Idealizing the Real World– But How?

- Goal: ‘abstract away’ as many non-essential properties as possible.
- Prove statements in simplified, idealized model.
- Intuition: If model is good, proofs are meaningful in the real world.

Notable Examples of Idealized Models

- Random Oracle Model (ROM): idealizes hash functions.

Notable Examples of Idealized Models

- Random Oracle Model (ROM): idealizes hash functions.
- Generic Group Model (GGM): idealizes cyclic groups.

Generic Group Algorithms

Let $\mathbb{G} = \langle G, \circ, g \rangle$. A is *generic*, if it only computes over $\mathbb{G}$ as follows:

§ Given $a, b \in \mathbb{G}$, compute $c := a \circ b$

Generic Group Algorithms

Let $\mathbb{G} = \langle G, \circ, g \rangle$. A is *generic*, if it only computes over $\mathbb{G}$ as follows:

- Given $a, b \in \mathbb{G}$, compute $c := a \circ b$
- Given $a, b \in \mathbb{G}$, check whether $a = b$.

Generic Group Algorithms: Pros

- Work in every cyclic group.

Generic Group Algorithms: Pros

- Work in every cyclic group.
- Information theoretic lower bounds (DLP, CDH, DDH etc.)

Generic Group Algorithms: Pros

- Work in every cyclic group.
- Information theoretic lower bounds (DLP, CDH, DDH etc.)
- Fitting abstraction for (some) *elliptic curves*.

Generic Group Algorithms: Cons

- Representation-based exploits: Computing Jacobi symbols, index calculus-based attacks, computing pairings etc.

Generic Group Algorithms: Cons

- Representation-based exploits: Computing Jacobi symbols, index calculus-based attacks, computing pairings etc.
- Deriving lower bounds is difficult/tedious.

Generic Group Algorithms: Cons

- Representation-based exploits: Computing Jacobi symbols, index calculus-based attacks, computing pairings etc.
- Deriving lower bounds is difficult/tedious.
- Lower bounds are not ‘modular’.

This Talk: Algebraic Group Model (AGM)

- Strictly weaker model assumptions than GGM.

This Talk: Algebraic Group Model (AGM)

- Strictly weaker model assumptions than GGM.
- GGM $\leq$ AGM $\leq$ Standard Model.

This Talk: Algebraic Group Model (AGM)

- Strictly weaker model assumptions than GGM.
- GGM $\leq$ AGM $\leq$ Standard Model.
- Lies in between GGM and standard model.

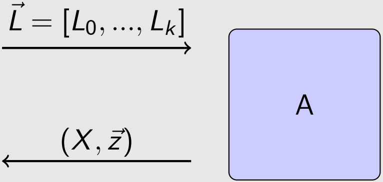
This Talk: Algebraic Group Model (AGM)

- Strictly weaker model assumptions than GGM.
- GGM $\leq$ AGM $\leq$ Standard Model.
- Lies in between GGM and standard model.
- Provides improved abstraction of reality over GGM.

This Talk: Algebraic Group Model (AGM)

- Strictly weaker model assumptions than GGM.
- GGM $\leq$ AGM $\leq$ Standard Model.
- Lies in between GGM and standard model.
- Provides improved abstraction of reality over GGM.
- Still allows easy proofs.

Algebraic Algorithms



- A takes as input list $\vec{L}$ of group elements.
- Outputs representation $\vec{z}$ of X, i.e., $X = \prod_i L_i^{z_i}$.

Algebraic Algorithms: Some Background

- First introduced by Paillier and Vergnaud in 05.

Algebraic Algorithms: Some Background

- First introduced by Paillier and Vergnaud in 05.
- *Algebraic reductions* mostly used for meta-reductions.

Algebraic Algorithms: Some Background

- First introduced by Paillier and Vergnaud in 05.
- *Algebraic reductions* mostly used for meta-reductions.
- Some exceptions: E.g. Abdalla et al. (S&P '15) consider an algebraic *adversary* in one of their proofs

New Idea: Algebraic Group Model

- All algorithms are modeled as algebraic, i.e., also adversaries in security experiments.

New Idea: Algebraic Group Model

- All algorithms are modeled as algebraic, i.e., also adversaries in security experiments.
- This gives *strictly weaker* model assumptions than the GGM.

Relation to the GGM

Lemma 1

Every generic algorithm is an algebraic algorithm.

- A generic algorithm A computes any output from elements in the list $\vec{L}$ via $\circ$.
- Thus, it must know a representation $\vec{z}$ for every output.
- We assume that it outputs $\vec{z}$ 'for free'.

Bounds for GGM via Reduction in AGM

Theorem 2 (Composition)

- *Suppose that:*

Bounds for GGM via Reduction in AGM

Theorem 2 (Composition)

- *Suppose that:*
  - $\texttt{true} \overset{\texttt{GGM}}{\Longrightarrow} S$

Bounds for GGM via Reduction in AGM

Theorem 2 (Composition)

- *Suppose that:*
  - $\texttt{true} \overset{\texttt{GGM}}{\Longrightarrow} S$
  - $S \overset{\texttt{AGM}}{\Longrightarrow} T$

Bounds for GGM via Reduction in AGM

Theorem 2 (Composition)

- *Suppose that:*
  - $\texttt{true} \overset{\texttt{GGM}}{\Longrightarrow} S$
  - $S \overset{\texttt{AGM}}{\Longrightarrow} T$
- *Then* $\texttt{true} \overset{\texttt{GGM}}{\Longrightarrow} T$*, if reduction in AGM is a generic algorithm.*

Proofs in AGM vs. Proofs in GGM

- GGM: Lower bounds for algorithms via combinatorics.
- AGM: Reductions.

Using the AGM

- How do we prove reductions in the AGM?
- Want to make use of representation vector $\vec{z}$.

Using the AGM: Example

- CDH: Given g, g^x, g^y, compute g^{xy}.
- DLP: Given g^u, compute u.
- **Lemma** : DLP $\overset{\texttt{AGM}}{\Leftrightarrow}$ CDH
- Breaking CDH algebraically is as hard as solving DLP.

CDH Game

Challenger: $U = g^u$ Adversary

$$\xrightarrow{g, g^x, g^y}$$

$$\xleftarrow{g^{xy}, z = (a, b, c)}$$

§ $g^{xy} = (g^x)^a (g^y)^b g^c$ is equivalent to $xy \equiv_p xa + yb + c$.

CDH Game

Challenger: $U = g^u$ Adversary

$$\xrightarrow{g, g^x, g^y}$$

$$\xleftarrow{g^{xy}, z = (a, b, c)}$$

- $g^{xy} = (g^x)^a (g^y)^b g^c$ is equivalent to $xy \equiv_p xa + yb + c$.
- $x \equiv_p \frac{yb+c}{y-a} \implies$: Can either solve for x or $a \equiv_p y$.

CDH Game

Challenger: $U = g^u$ Adversary

$$\xrightarrow{\quad g,\, g^x,\, g^y \quad}$$

$$\xleftarrow{\quad g^{xy},\, z = (a, b, c) \quad}$$

- $g^{xy} = (g^x)^a (g^y)^b g^c$ is equivalent to $xy \equiv_p xa + yb + c$.
- $x \equiv_p \frac{yb+c}{y-a} \implies$: Can either solve for x or $a \equiv_p y$.
- Idea: Set $U = g^x$ OR $U = g^y$ and choose the other randomly.

CDH Game

Challenger: $U = g^u$ Adversary

$$\xrightarrow{g, g^x, g^y}$$

$$\xleftarrow{g^{xy}, z = (a, b, c)}$$

- $g^{xy} = (g^x)^a (g^y)^b g^c$ is equivalent to $xy \equiv_p xa + yb + c$.
- $x \equiv_p \frac{yb+c}{y-a} \implies$: Can either solve for x or $a \equiv_p y$.
- Idea: Set $U = g^x$ OR $U = g^y$ and choose the other randomly.
- Succeeds with probability $1/2$.

Strong DH

- Strong DH (SDH): CDH assumption with additional DDH oracle to ‘fixed base’ g^x.

Strong DH

- Strong DH (SDH): CDH assumption with additional DDH oracle to 'fixed base' g^x.
- Related to Gap DH (but slightly stronger).

Strong DH

- Strong DH (SDH): CDH assumption with additional DDH oracle to ‘fixed base’ g^x.
- Related to Gap DH (but slightly stronger).
- Used for DHIES, Hashed ElGamal etc.

Strong DH

- Strong DH (SDH): CDH assumption with additional DDH oracle to 'fixed base' g^x.
- Related to Gap DH (but slightly stronger).
- Used for DHIES, Hashed ElGamal etc.
- **Lemma** : DLP $\overset{\texttt{AGM}}{\Leftrightarrow}$ SDH

Equivalence to DLP in AGM: LRSW

Challenger		Adversary
Oracle	$\xrightarrow{(X = g^x, Y = g^y)}$	
	$\xleftarrow{m_i}$	$\updownarrow$ Q times
	$\xrightarrow{(a_i, a_i^y, a_i^{xym_i+x})}$	
	$\xleftarrow{m, (a, a^y, a^{xym+x})}$	

- Basis of Camenisch-Lysyanskaya signature.
- Used for RFID Tags, Anonymous Credentials, etc.
- **Lemma** : DLP $\overset{\mathtt{AGM}}{\Leftrightarrow}$ LRSW

Some more Results

§ *Tight* reduction of BLS (short, pairing based signature) to DLP.

Some more Results

- *Tight* reduction of BLS (short, pairing based signature) to DLP.
- q-type variant of DDH $\overset{\texttt{AGM}}{\Longrightarrow}$ ElGamal CCA1

Some more Results

- *Tight* reduction of BLS (short, pairing based signature) to DLP.
- q-type variant of DDH $\stackrel{\mathtt{AGM}}{\Longrightarrow}$ ElGamal CCA1
- q-type variant of DLP $\stackrel{\mathtt{AGM}}{\Longrightarrow}$ Groth's ZK-SNARK (EC16)

Other Candidates

- AGM is ideal for analyzing group based crypto systems that would otherwise be analyzed in GGM.

Other Candidates

- AGM is ideal for analyzing group based crypto systems that would otherwise be analyzed in GGM.
- Examples: Structure preserving signatures, ZK-SNARKS.

Summary

- AGM is in between GGM and Standard Model.

Summary

- AGM is in between GGM and Standard Model.
- Reduction based, easy to work with.

Summary

- AGM is in between GGM and Standard Model.
- Reduction based, easy to work with.
- Captures a broad class of important algorithms.

Summary

- AGM is in between GGM and Standard Model.
- Reduction based, easy to work with.
- Captures a broad class of important algorithms.
- Circumvents impossibility results for black box reductions.

Summary

- AGM is in between GGM and Standard Model.
- Reduction based, easy to work with.
- Captures a broad class of important algorithms.
- Circumvents impossibility results for black box reductions.
- Results from AGM carry over to GGM.

Open Questions

§ Meta-theorems to cover broad class of assumptions?

Open Questions

- Meta-theorems to cover broad class of assumptions?
- Automated proof tools in AGM?

Open Questions

- Meta-theorems to cover broad class of assumptions?
- Automated proof tools in AGM?
- Possibility results along the lines of Dent (Asiacrypt '02)?

Open Questions

- Meta-theorems to cover broad class of assumptions?
- Automated proof tools in AGM?
- Possibility results along the lines of Dent (Asiacrypt '02)?
- Proofs for composite order groups?

Open Questions

- Meta-theorems to cover broad class of assumptions?
- Automated proof tools in AGM?
- Possibility results along the lines of Dent (Asiacrypt '02)?
- Proofs for composite order groups?
- Extend to Decisional Assumptions.

RUB

RUHR-UNIVERSITÄT BOCHUM

Many thanks for your attention!

QUESTIONS?

hgi
Horst Görtz Institut
für IT-Sicherheit