(Future) Relationship between Topology and Cryptography

Koji NUIDA

IMI, Kyushu University

TopoICComp2025 @Fukuoka 2025.12.5

Self Introduction

- Research interests: Combinatorial group theory (Coxeter groups), then Cryptography (& Math)
  - Ph.D. thesis on the isomorphism problem for Coxeter groups
- Gave a talk at this conference series in 9 years ago (Oct. 29, 2016)
  - “How to Apply Topology to Cryptology, Hopefully”
  - When I was working at AIST
  - Some (many?) overlaps with today’s talk

Advertising My Work in Math

- Coxeter group
  $W = \langle S \mid s^2\ (\forall s \in S), (st)^{m(s,t)}\ (\forall s \neq t \in S) \rangle$
  - S: Coxeter generating set
- Sometimes W has two (or more) non-conjugate Coxeter generating sets S (with possibly different $m(s,t)$'s)
  - E.g., $W(A_1 \times A_2) \simeq W(I_2(6))$
- Def.: W is **strongly rigid** if S is unique up to conjugation

- [Charney–Davis 2000]: A (large?) **topologically** defined class of strongly rigid Coxeter groups
- [Howlett–Mühlherr–N. 2018]: Complete (combinatorial) characterization of strongly rigid Coxeter groups of finite ranks
- [Mühlherr–N. 2021]: A (large) class of strongly rigid Coxeter groups of infinite ranks

Contents

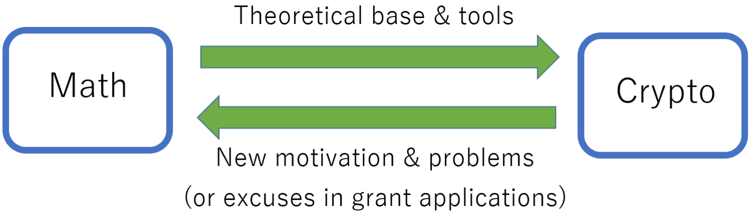
- Relations between Math & Crypto
- Topic 1: Zero-Knowledge Proofs
- Topic 2: Fully Homomorphic Encryption

Contents

- Relations between Math & Crypto
- Topic 1: Zero-Knowledge Proofs
- Topic 2: Fully Homomorphic Encryption

- Methods of ensuring that information technology can be used as expected, even if some people may try to obstruct such use
  - "Some people": **adversary** (or **attacker**)
- Examples: encryption (wants to keep data secret), digital signature (wants to ensure that messages are from a true sender)
- Usually requiring secret information which adversaries do not know
  - Some exception exists (e.g., cryptographic hash functions)

Relations between Math & Crypto



Relations between Math & Crypto: Examples

- Number theory: RSA cryptosystem
- Algebraic geometry: elliptic curve crypto, pairing-based crypto, isogeny-based crypto
- (Noisy) linear algebra: code-based crypto, lattice-based crypto
- Graph theory: cryptographic hash functions
- Gröbner basis: multivariate crypto

Relations between Math & Crypto: Examples

- (Not algebraic) geometry:

Relations between Math & Crypto: Examples

- (Not algebraic) geometry:
  - Shamir's secret sharing (any two points determine a unique line) [Shamir 1979]

Relations between Math & Crypto: Examples

- (Not algebraic) geometry:
  - Shamir's secret sharing (any two points determine a unique line) [Shamir 1979]
  - Algebraic surface cryptosystem (difficulty of section finding) [Akiyama–Goto–Miyake 2009]

- (Not algebraic) geometry:
  - Shamir's secret sharing (any two points determine a unique line) [Shamir 1979]
  - Algebraic surface cryptosystem (difficulty of section finding) [Akiyama–Goto–Miyake 2009]
  - Attempt of applying Finsler geometry to crypto [Nagano–Anada 2023]

- (Not algebraic) geometry:
  - Shamir's secret sharing (any two points determine a unique line) [Shamir 1979]
  - Algebraic surface cryptosystem (difficulty of section finding) [Akiyama–Goto–Miyake 2009]
  - Attempt of applying Finsler geometry to crypto [Nagano–Anada 2023]
  - ...

Relations between Math & Crypto: Examples

- (Not algebraic) geometry:
  - Shamir's secret sharing (any two points determine a unique line) [Shamir 1979]
  - Algebraic surface cryptosystem (difficulty of section finding) [Akiyama–Goto–Miyake 2009]
  - Attempt of applying Finsler geometry to crypto [Nagano–Anada 2023]
  - ...
- **Topology**: ??

Crypto with Topology-Related Groups

- Key exchange protocol using **braid groups** [Ko et al. 2000]
  - cf. [Anshel–Anshel–Goldfeld 1999]
- Most famous example of "group-based crypto"
- But broken by [Myasnikov–Shpilrain–Ushakov 2005] etc.
- See e.g., [Garber, arXiv:0711.3941] for a survey on "braid group crypto"

Contents

- Situation: A player P knows a solution of a problem, and wants to convince a player V that P certainly knows a solution

Zero-Knowledge Proofs: Introduction

- Situation: A player P knows a solution of a problem, and wants to convince a player V that P certainly knows a solution
    - But V is so distrustful, needs a “proof”

- Situation: A player P knows a solution of a problem, and wants to convince a player V that P certainly knows a solution
  - But V is so distrustful, needs a “proof”
  - And the solution is valuable, so P cannot tell it to V

- Situation: A player P knows a solution of a problem, and wants to convince a player V that P certainly knows a solution
  - But V is so distrustful, needs a “proof”
  - And the solution is valuable, so P cannot tell it to V
- No way?

- [Goldwasser–Micali–Rackoff 1985]: Such a "proof without telling the solution" (**zero-knowledge proof**, ZKP) is possible for a class of problems

- [Goldwasser–Micali–Rackoff 1985]: Such a "proof without telling the solution" (**zero-knowledge proof**, ZKP) is possible for a class of problems
  - In fact, for any NP language
  - Intuitively, an NP language is a problem for which validity check of a given solution is easy

Zero-Knowledge Proofs: Example

Zero-Knowledge Proofs: Example

- P's solution: an isomorphism φ between two structures G_0, G_1 (e.g., graphs)

- P's solution: an isomorphism φ between two structures G_0, G_1 (e.g., graphs)
- Protocol:

Zero-Knowledge Proofs: Example

- P's solution: an isomorphism φ between two structures G_0, G_1 (e.g., graphs)
- Protocol:
  - P tells V a new H isomorphic to G_0

- P's solution: an isomorphism φ between two structures G_0, G_1 (e.g., graphs)
- Protocol:
  - P tells V a new H isomorphic to G_0
  - V tells P a random $b \in \{0, 1\}$

- P's solution: an isomorphism φ between two structures G_0, G_1 (e.g., graphs)
- Protocol:
  - P tells V a new H isomorphic to G_0
  - V tells P a random $b \in \{0, 1\}$
  - P tells V an isomorphism $G_b \to H$ (and V checks if it is an isomorphism)

Zero-Knowledge Proofs: Example

- P's solution: an isomorphism φ between two structures G_0, G_1 (e.g., graphs)
- Protocol:
  - P tells V a new H isomorphic to G_0
  - V tells P a random $b \in \{0, 1\}$
  - P tells V an isomorphism $G_b \to H$ (and V checks if it is an isomorphism)
  - Repeat it until V is convinced

- Soundness: A true P can always answer

Zero-Knowledge Proofs: Example

- Soundness: A true P can always answer
- Completeness: W.h.p., a fake P cannot always answer

Zero-Knowledge Proofs: Example

- Soundness: A true P can always answer
- Completeness: W.h.p., a fake P cannot always answer
- Zero-knowledge (informal): Only a single isomorphism $G_b \to H$ gives no information on the original φ
  - Can be stated and proved rigorously

Zero-Knowledge Proofs: Example

- Soundness: A true P can always answer
- Completeness: W.h.p., a fake P cannot always answer
- Zero-knowledge (informal): Only a single isomorphism $G_b \to H$ gives no information on the original φ
    - Can be stated and proved rigorously

- You can try it with topological structures such as manifolds, knots, etc.

Card-Based Zero-Knowledge Proofs

- A too distrustful V may even distrust the computer on which the protocol is executed
- A possible direction: **Card-based** ZKP
  - **Card-based crypto** [den Boer 1989]: Doing crypto by physical cards with open/face-down operations, permutations, shuffle operations, etc.
  - Motivated by visible demonstration, recreation, education (and more!)

Example: Card-Based ZKP for Sudoku

- Sudoku: A puzzle to put numbers $1, 2, \ldots, 9$ in the cells on a 9×9 board to satisfy:
  - Each row has different numbers
  - Each column has different numbers
  - Each of the nine 3×3 sub-boards has different numbers
  - Consistent with the initial (partial) placement

Example: Card-Based ZKP for Sudoku

- A ZKP protocol [Gradwohl et al. 2007] for solution of Sudoku:

Example: Card-Based ZKP for Sudoku

- A ZKP protocol [Gradwohl et al. 2007] for solution of Sudoku:
  - P puts face-down number cards on the board (with consistency) by following the solution

- A ZKP protocol [Gradwohl et al. 2007] for solution of Sudoku:
  - P puts face-down number cards on the board (with consistency) by following the solution
  - V verifies one of the row, column, and sub-board conditions by shuffling the 9 cards and opening them

Example: Card-Based ZKP for Sudoku

- A ZKP protocol [Gradwohl et al. 2007] for solution of Sudoku:
    - P puts face-down number cards on the board (with consistency) by following the solution
    - V verifies one of the row, column, and sub-board conditions by shuffling the 9 cards and opening them
    - Repeat it until V is convinced

Example: Card-Based ZKP for Sudoku

- A ZKP protocol [Gradwohl et al. 2007] for solution of Sudoku:
  - P puts face-down number cards on the board (with consistency) by following the solution
  - V verifies one of the row, column, and sub-board conditions by shuffling the 9 cards and opening them
  - Repeat it until V is convinced
- For true P, the opened cards are always a permutation of $1, 2, \ldots, 9$ ($\leadsto$ no information)

- A ZKP protocol [Gradwohl et al. 2007] for solution of Sudoku:
  - P puts face-down number cards on the board (with consistency) by following the solution
  - V verifies one of the row, column, and sub-board conditions by shuffling the 9 cards and opening them
  - Repeat it until V is convinced
- For true P, the opened cards are always a permutation of $1, 2, \ldots, 9$ ($\rightsquigarrow$ no information)
- For fake P, it is revealed when V selects an unsatisfied condition (with positive probability)

ZKP for Rubik's Cube

- ZKP for solutions of Rubik's Cube is also possible with
  - Computer (via the general feasibility result)
  - Cards [Kimura–Mizuki–Komano 2024 (in Japanese)]
  - Rubik's Cube itself [Kimura–Mizuki 2024 (in Japanese)]

ZKP for/by ??

- Ordinary or "physical" ZKP for topology-related problems?
  - E.g., solution of Teruaki puzzle[1]?
- There are physical crypto using various tools: cards (of various types/shapes), coins, balances, polarizing plates, PEZs, etc.
  - Physical crypto using topology-related objects, e.g., Möbius Kaleidocycles[2]?

[1] https://w.atwiki.jp/kazushiahara/pages/32.html

[2] https://www.kyushu-u.ac.jp/ja/researches/view/908

Contents

- Relations between Math & Crypto
- Topic 1: Zero-Knowledge Proofs
- Topic 2: Fully Homomorphic Encryption

Warm-Up: Diffie–Hellman Key Exchange

- Encrypted communication using secret-key encryption requires a secret key shared by the sender and receiver in advance
- How to securely share the secret key without encryption?
- The earliest solution: **Diffie–Hellman** (DH) **key exchange** [Diffie–Hellman 1976]

Protocol between parties P_1 and P_2:

1.
2.
3.

Getting a common (random) secret element:

with no pre-shared secret

Protocol between parties P_1 and P_2:

1. Choose a finite cyclic group $G = \langle g \rangle$ in public
2.
3.

Getting a common (random) secret element:

with no pre-shared secret

Warm-Up: Diffie–Hellman Key Exchange

Protocol between parties P_1 and P_2:

1. Choose a finite cyclic group $G = \langle g \rangle$ in public
2. P_i sends $h_i := g^{a_i}$ to P_{3-i}, while hiding $a_i \in \mathbb{Z}$
3.

Getting a common (random) secret element:

with no pre-shared secret

Protocol between parties P_1 and P_2:

1. Choose a finite cyclic group $G = \langle g \rangle$ in public
2. P_i sends $h_i := g^{a_i}$ to P_{3-i}, while hiding $a_i \in \mathbb{Z}$
3. Given h_{3-i}, P_i computes $K_i := {h_{3-i}}^{a_i}$

Getting a common (random) secret element:

with no pre-shared secret

Warm-Up: Diffie–Hellman Key Exchange

Protocol between parties P_1 and P_2:

1. Choose a finite cyclic group $G = \langle g \rangle$ in public
2. P_i sends $h_i := g^{a_i}$ to P_{3-i}, while hiding $a_i \in \mathbb{Z}$
3. Given h_{3-i}, P_i computes $K_i := {h_{3-i}}^{a_i}$

Getting a common (random) secret element:

$$K_1 = (g^{a_2})^{a_1} = g^{a_2 a_1} = g^{a_1 a_2} = (g^{a_1})^{a_2} = K_2$$

with no pre-shared secret

Warm-Up: Diffie–Hellman Key Exchange

Public: $G = \langle g \rangle$ and $h_i \in G$

Secret: a_i with $h_i = g^{a_i}$

Public: $G = \langle g \rangle$ and $h_i \in G$
Secret: a_i with $h_i = g^{a_i}$

$\Rightarrow$ The **discrete logarithm** (DL) **problem** in G must be computationally hard:

(DL) $\boxed{\text{Given } g, h, \text{ find an } x \text{ with } h = g^x \text{ in } G}$

Public: $G = \langle g \rangle$ and $h_i \in G$
Secret: a_i with $h_i = g^{a_i}$

$\Rightarrow$ The **discrete logarithm** (DL) **problem** in G must be computationally hard:

(DL) $\boxed{\text{Given } g, h, \text{ find an } x \text{ with } h = g^x \text{ in } G}$

- DL over $\mathbb{Z}/n\mathbb{Z}$ is easy, DL over ${\mathbb{F}_p}^{\times}$ is fairly hard, and DL over elliptic curve groups is much harder

Warm-Up: Diffie–Hellman Key Exchange

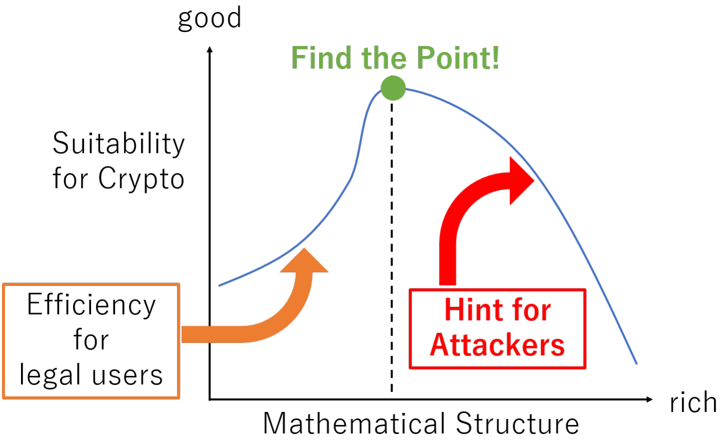
Public: $G = \langle g \rangle$ and $h_i \in G$
Secret: a_i with $h_i = g^{a_i}$

$\Rightarrow$ The **discrete logarithm** (DL) **problem** in G must be computationally hard:

(DL) $\boxed{\text{Given } g, h, \text{ find an } x \text{ with } h = g^x \text{ in } G}$

- DL over $\mathbb{Z}/n\mathbb{Z}$ is easy, DL over $\mathbb{F}_p{}^\times$ is fairly hard, and DL over elliptic curve groups is much harder
  - Even though the groups are isomorphic
  - Because elliptic curve groups have “less structure” than other groups

A New Viewpoint from Cryptography



(Mathematician: more structures, more happiness)

Fully Homomorphic Encryption (FHE)

(over plaintext space $\mathbb{F}_2 = \{0, 1\}$)

Fully Homomorphic Encryption (FHE)

(over plaintext space $\mathbb{F}_2 = \{0, 1\}$)

- $\mathsf{Decrypt} \colon \{\text{ciphertexts}\} \to \{\text{plaintexts}\}$

Fully Homomorphic Encryption (FHE)

(over plaintext space $\mathbb{F}_2 = \{0, 1\}$)

- $\mathrm{Decrypt} \colon \{\text{ciphertexts}\} \to \{\text{plaintexts}\}$
- $\exists\ \widetilde{\mathrm{add}}, \widetilde{\mathrm{mult}}$ on ciphertexts s.t. Decrypt is "ring-homomorphic":
  - $\mathrm{Decrypt}(\widetilde{\mathrm{add}}(c_1, c_2))$
    $= \mathrm{Decrypt}(c_1) + \mathrm{Decrypt}(c_2)$
  - $\mathrm{Decrypt}(\widetilde{\mathrm{mult}}(c_1, c_2))$
    $= \mathrm{Decrypt}(c_1) \cdot \mathrm{Decrypt}(c_2)$

(over plaintext space $\mathbb{F}_2 = \{0, 1\}$)

- Decrypt: $\{\text{ciphertexts}\} \to \{\text{plaintexts}\}$
- $\exists$ $\widetilde{\text{add}}, \widetilde{\text{mult}}$ on ciphertexts s.t. Decrypt is "ring-homomorphic":
  - $\text{Decrypt}(\widetilde{\text{add}}(c_1, c_2)) = \text{Decrypt}(c_1) + \text{Decrypt}(c_2)$
  - $\text{Decrypt}(\widetilde{\text{mult}}(c_1, c_2)) = \text{Decrypt}(c_1) \cdot \text{Decrypt}(c_2)$
- "Homomorphic" computation over encrypted data
- First construction: [Gentry 2009]

(Too) Simplified Example [van Dijk et al. 2010]

Choose prime p and integer N s.t. $p \mid N$

$\mathrm{Encrypt}(m) = p\alpha + 2r + m \bmod N$ for $m \in \mathbb{F}_2$

(Too) Simplified Example [van Dijk et al. 2010]

Choose prime p and integer N s.t. $p \mid N$

$\mathrm{Encrypt}(m) = p\alpha + 2r + m \bmod N$ for $m \in \mathbb{F}_2$

$\mathrm{Decrypt}(c) = (c \bmod p) \bmod 2$

- It works iff r is "not too large"

(Too) Simplified Example [van Dijk et al. 2010]

Choose prime p and integer N s.t. $p \mid N$

$\mathrm{Encrypt}(m) = p\alpha + 2r + m \bmod N$ for $m \in \mathbb{F}_2$

$\mathrm{Decrypt}(c) = (c \bmod p) \bmod 2$

- It works iff r is "not too large"

$\widetilde{\mathrm{add}} = +$, $\widetilde{\mathrm{mult}} = \cdot$ in $\mathbb{Z}/N\mathbb{Z}$

- But iteration of operations is limited (r grows)

(Too) Simplified Example [van Dijk et al. 2010]

Choose prime p and integer N s.t. $p \mid N$

$\mathrm{Encrypt}(m) = p\alpha + 2r + m \bmod N$ for $m \in \mathbb{F}_2$

$\mathrm{Decrypt}(c) = (c \bmod p) \bmod 2$

- It works iff r is "not too large"

$\widetilde{\mathrm{add}} = +$, $\widetilde{\mathrm{mult}} = \cdot$ in $\mathbb{Z}/N\mathbb{Z}$

- But iteration of operations is limited (r grows)

"Bootstrapping" can reset the "noise" r

- But very inefficient so far
- Common to other FHE schemes

Towards FHE without Bootstrapping

A (hopefully) possible strategy ([N. 2021], etc.):

A (hopefully) possible strategy ([N. 2021], etc.):

1. "Embed" $\mathbb{F}_2$ into a finite (non-commutative) group G
   - Operations of $\mathbb{F}_2$ realized by operations of G

A (hopefully) possible strategy ([N. 2021], etc.):

1. "Embed" $\mathbb{F}_2$ into a finite (non-commutative) group G
   - Operations of $\mathbb{F}_2$ realized by operations of G
2. Take a surjective group hom. $\pi\colon \widetilde{G} \to G$ with some finite group $\widetilde{G}$ s.t.:
   - Elements of $\ker(\pi)$ can be efficiently sampled (with some public information)
   - π is hard-to-compute without secret key
   - π is easy-to-compute with secret key

A (hopefully) possible strategy ([N. 2021], etc.):

1. "Embed" $\mathbb{F}_2$ into a finite (non-commutative) group G
   - Operations of $\mathbb{F}_2$ realized by operations of G
2. Take a surjective group hom. $\pi\colon \widetilde{G} \to G$ with some finite group $\widetilde{G}$ s.t.:
   - Elements of $\ker(\pi)$ can be efficiently sampled (with some public information)
   - π is hard-to-compute without secret key
   - π is easy-to-compute with secret key

Ciphertexts: elements of $\widetilde{G}$, Decrypt $= \pi$

[Guillot et al., arXiv:2510.21483]

- $0 \mapsto \sigma_0 := \mathrm{id} \in S_6$, $1 \mapsto \sigma_1 := (15)(34)$
- $\mathrm{add}'(x, y) := xy$
- $\mathrm{mult}'(x, y) := a_1 x a_1 a_2 y a_2 a_1 x a_1 a_2 y a_2$ where $a_1 := (12)(56)$, $a_2 := (35)$
  - Written w.r.t. action from the right (i.e., left-side elements act firstly)
- Then $\mathrm{add}'(\sigma_{b_1}, \sigma_{b_2}) = \sigma_{b_1 + b_2}$, $\mathrm{mult}'(\sigma_{b_1}, \sigma_{b_2}) = \sigma_{b_1 \cdot b_2}$

A Recent Approach

- It looks so difficult to find a suitable group hom. $\pi\colon \widetilde{G} \to G$

A Recent Approach

- It looks so difficult to find a suitable group hom. $\pi\colon \widetilde{G} \to G$
- Approach by [Guillot et al., arXiv:2510.21483]:
  - Take some $\widetilde{G}$
  - Take an ambient group $\widetilde{G}_0 \supseteq \widetilde{G}$
  - Then publish a (non-confluent and "pseudo-bounded") **rewriting system** $\mathscr{G}$ for group presentation of $\widetilde{G}_0$
  - Every computation (except for Decrypt) is done over $\mathscr{G}$, without explicit structure of $\widetilde{G}_0$

Open Problems:

- Concrete construction (rather than rewriting system) of a suitable group hom. $\widetilde{G} \to S_n$ ($n \geq 5$), associated to some topological object? (Cf. elliptic curve groups for DH key exchange)
- Embedding of $\mathbb{F}_2$ into other topology-related objects?
(E.g., quandles from knot theory?)