Cryptography, Information Security, and Mathematics: Recent Advances

縫田　光司（Koji NUIDA）

産業技術総合研究所 (AIST)
/ JST さきがけ（JST PRESTO）

情報セキュリティにおける数学的方法とその実践
2016 年 12 月 19 日　@北海道大学

For SCIS People

- The (extended) deadline for paper submission has passed (almost now, at 10:00)
- Please take a rest before the next talk :-)

A Few Years Ago (Maybe)

- (Rough) scenario of a police TV drama:
  - Victim was a mathematician, he found **"ultimate property of primes"**
  - He decided to delete his result, because **it will cause "fatal catastrophe" about cryptography in the whole world**
  - (and was killed by his friend who disagreed)

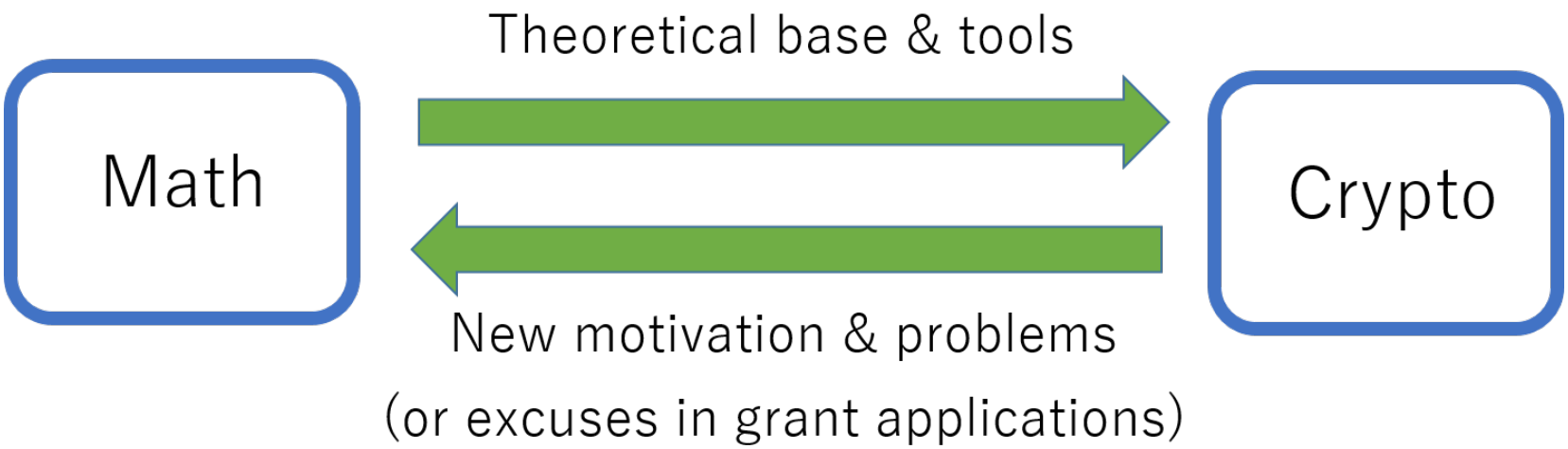
A Few Years Ago (Maybe)

- (Rough) scenario of a police TV drama:
  - Victim was a mathematician, he found **"ultimate property of primes"**
  - He decided to delete his result, because **it will cause "fatal catastrophe" about cryptography in the whole world**
  - (and was killed by his friend who disagreed)
- The "catastrophe" is fictional, but this example shows: It has become recognized so widely that primes and cryptography are related (in some unknown way)

Main Theme of This Talk

- The RSA cryptosystem would have made relation of math. and crypto. so popular, but ...
- **RSA is NOT the only such relation**
  - (and "RSA and elliptic curve crypto." are not the only, too)

Relation between Mathematics and Cryptography



暗号及び情報セキュリティと数学の相関
ワークショップ (workshop on interaction between CRyptography, Information Security and MATHematics)

- Invited talks on math. & crypto. (and more)
- **This year's: Dec. 26th @Tokyo (Odaiba)**
  - Organizers: K. Nuida, T. Abe, S. Kaji, H. Kurihara, T. Maeno, Y. Numata
    - (Underlined: Speakers of the current workshop)

linear-algebraic technique in integer factorization; dynamical systems and sensor networks (Z. Arai); algorithmic randomness and quantum key distribution; network codings and sheaf cohomology; algorithmic randomness and random oracles; algebraic-geometrical approach on LLL algorithm; NP vs. P problem; algebraic surfaces and cryptography (K. Akiyama); cryptography from group theory (N.); group rings and NTRU cryptosystem; quantum computation; isogenies for elliptic curves (K. Takashima); Bernoulli numbers and use of FHE (N.)

CRISMATH Workshop: This Year's Talks

- 川合 豊「鍵が固定された場合の暗号方式の安全性について」
- 品川 和雅「カードとシャッフルから見るカード暗号プロトコル」
- 白勢 政明「楕円曲線，暗号，素因数分解」
  - On applications of elliptic curves to integer factorization
- Taechan Kim「Extended Tower Number Field Sieve」
  - On speed-up for discrete logarithms
- 縫田 光司「秘密計算で他者に迷惑をかけずに疑似乱数を使えるか」

Public Key Encryption (PKE)

- To conceal messages from attackers

Public Key Encryption (PKE)

- To conceal messages from attackers
- Encryption: message $\mapsto$ ciphertext
  - using **public** encryption key pk

Public Key Encryption (PKE)

- To conceal messages from attackers
- Encryption: message $\mapsto$ ciphertext
  - using **public** encryption key pk
- Decryption: ciphertext $\mapsto$ message
  - using **secret** decryption key sk

Public Key Encryption (PKE)

- To conceal messages from attackers
- Encryption: message $\mapsto$ ciphertext
  - using **public** encryption key pk
- Decryption: ciphertext $\mapsto$ message
  - using **secret** decryption key sk
- $\mathrm{Dec}_{\mathrm{sk}}(\mathrm{Enc}_{\mathrm{pk}}(m)) = m$

Public Key Encryption (PKE)

- To conceal messages from attackers
- Encryption: message $\mapsto$ ciphertext
  - using **public** encryption key pk
- Decryption: ciphertext $\mapsto$ message
  - using **secret** decryption key sk
- $\mathrm{Dec}_{\mathrm{sk}}(\mathrm{Enc}_{\mathrm{pk}}(m)) = m$
- pk **should not yield information on** sk

Before the RSA: "One-Time Pad"

- **Symmetric** key encryption, where pk $=$ sk should be hidden

Before the RSA: "One-Time Pad"

- **Symmetric** key encryption, where pk $=$ sk should be hidden

Given "message" $m \in M$ (*finite* additive group)

- Originally, $M = (\mathbb{F}_2)^n$ (bitwise XOR)

Before the RSA: "One-Time Pad"

- **Symmetric** key encryption, where pk $=$ sk should be hidden

Given "message" $m \in M$ (*finite* additive group)

- Originally, $M = (\mathbb{F}_2)^n$ (bitwise XOR)

If pk $= k \in M$ is uniformly random, then ciphertext $c = \mathrm{Enc}_k(m) := m + k$ is **independent of** m

Before the RSA: "One-Time Pad"

- **Symmetric** key encryption, where pk $=$ sk should be hidden

Given "message" $m \in M$ (*finite* additive group)

- Originally, $M = (\mathbb{F}_2)^n$ (bitwise XOR)

If pk $= k \in M$ is uniformly random, then ciphertext $c = \mathrm{Enc}_k(m) := m + k$ is **independent of** m

- Perfectly hiding, **if** k **is used only once**

The RSA Cryptosystem [1977 ?]

- $N = pq$ (distinct primes)
- e, d with $ed \equiv 1 \pmod{(p-1)(q-1)}$

Given message $m \in (\mathbb{Z}/N\mathbb{Z})^\times$,

- $\mathrm{Enc}(m) := m^e$ (public key: (N, e))
- $\mathrm{Dec}(c) := c^d$ (secret key: d)

The RSA Cryptosystem [1977 ?]

- $N = pq$ (distinct primes)
- e, d with $ed \equiv 1 \pmod{(p-1)(q-1)}$

Given message $m \in (\mathbb{Z}/N\mathbb{Z})^\times$,

- $\mathrm{Enc}(m) := m^e$ (public key: (N, e))
- $\mathrm{Dec}(c) := c^d$ (secret key: d)

d would be computable if p, q were known

The RSA Cryptosystem [1977 ?]

- $N = pq$ (distinct primes)
- e, d with $ed \equiv 1 \pmod{(p-1)(q-1)}$

Given message $m \in (\mathbb{Z}/N\mathbb{Z})^\times$,

- $\mathrm{Enc}(m) := m^e$ (public key: (N, e))
- $\mathrm{Dec}(c) := c^d$ (secret key: d)

d would be computable if p, q were known

Drawback: Enc is deterministic ("textbook RSA")

- Improved variant is practically used

Computational Assumptions

In PKE, secret should not be found in "practical" (theoretically, probabilistic polynomial) time

- E.g. "Factoring N is hard" for the RSA

Computational Assumptions

In PKE, secret should not be found in "practical" (theoretically, probabilistic polynomial) time

- E.g. "Factoring N is hard" for the RSA
- Theoretically, just "assumption" (cf. P vs NP)
  - Practically, evaluated by experiments
  - Consensus: "(General) Number Field Sieve" would factorize $N \approx 2^{1024}$ in near future

The Number Field Sieve [Pollard+1990 ?]

State-of-the-art method to factoring integers
($\Longrightarrow$ to break the RSA cryptosystem)

- Using number fields
- Complexity: sub-exponential (not polynomial)
- Current status: Recommended to finish using RSA with $N \approx 2^{1024}$ (1024-bit key)

(Special) Number Field Sieve (1/3)

Choose

- $f(t) \in \mathbb{Z}[t]$, irreducible over $\mathbb{Q}$
- $\alpha \in \mathbb{C}$, with $f(\alpha) = 0$
- $m \in \mathbb{Z}$, with $f(m) \equiv 0 \pmod{N}$, $0 < m \ll N$

and assume the followings: (avoidable, in general)

- $K = \mathbb{Q}(\alpha)$ has integer ring $\mathcal{O}_K = \mathbb{Z}[\alpha]$
- $\mathcal{O}_K$ is UFD

(Special) Number Field Sieve (2/3)

$f(\alpha) = 0$, $f(m) \equiv 0 \pmod{N}$, $m \ll N$

Define $\varphi\colon \mathbb{Z}[\alpha] \xrightarrow{\text{hom.}} \mathbb{Z}/N\mathbb{Z}$, $\varphi(\alpha) = m$

(Special) Number Field Sieve (2/3)

$f(\alpha) = 0$, $f(m) \equiv 0 \pmod{N}$, $m \ll N$

Define $\varphi\colon \mathbb{Z}[\alpha] \xrightarrow{\text{hom.}} \mathbb{Z}/N\mathbb{Z}$, $\varphi(\alpha) = m$

For $a, b \in \mathbb{Z}$, compute $\varphi(a + b\alpha)$ in two ways:

- φ(factorization of $a + b\alpha$ in $\mathbb{Z}[\alpha]$)
- factorization of $\varphi(a + b\alpha)$ in $\mathbb{Z}$ ($\Rightarrow$ in $\mathbb{Z}/N\mathbb{Z}$)

Then get: ($\prod$ of primes) $\equiv$ ($\prod$ of primes) in $\mathbb{Z}/N\mathbb{Z}$

(Special) Number Field Sieve (3/3)

Relations ($\prod$ of primes) $\equiv$ ($\prod$ of primes) in $\mathbb{Z}/N\mathbb{Z}$ for each $a, b \in \mathbb{Z}$

(Special) Number Field Sieve (3/3)

Relations ($\prod$ of primes) $\equiv$ ($\prod$ of primes) in $\mathbb{Z}/N\mathbb{Z}$ for each $a, b \in \mathbb{Z}$

Combine them (by observing exponents) to get

$$x^2 \equiv y^2 \pmod{N}$$

yielding (with high probability)

$$1 < \gcd(x+y, N) < N \quad \text{or} \quad 1 < \gcd(x-y, N) < N$$

Prior to RSA — Diffie–Hellman Key Exchange [1976]

Protocol between parties P_1 and P_2

1.
2.

Getting a common (random) **secret element**

with **no pre-shared secret**

Prior to RSA — Diffie–Hellman Key Exchange [1976]

Protocol between parties P_1 and P_2

Choose $G = \langle g \rangle$ (finite cyclic) in public, then

1.

2.

Getting a common (random) **secret element**

with **no pre-shared secret**

Prior to RSA — Diffie–Hellman Key Exchange [1976]

Protocol between parties P_1 and P_2

Choose $G = \langle g \rangle$ (finite cyclic) in public, then

1. P_i sends $h_i := g^{a_i}$, while hiding $a_i \in \mathbb{Z}$
2.

Getting a common (random) **secret element**

with **no pre-shared secret**

Prior to RSA — Diffie–Hellman Key Exchange [1976]

Protocol between parties P_1 and P_2

Choose $G = \langle g \rangle$ (finite cyclic) in public, then

1. P_i sends $h_i := g^{a_i}$, while hiding $a_i \in \mathbb{Z}$
2. 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**

Prior to RSA — Diffie–Hellman Key Exchange [1976]

Protocol between parties P_1 and P_2

Choose $G = \langle g \rangle$ (finite cyclic) in public, then

1. P_i sends $h_i := g^{a_i}$, while hiding $a_i \in \mathbb{Z}$
2. 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**

Prior to RSA — Diffie–Hellman Key Exchange [1976]

Protocol between parties P_1 and P_2

Choose $G = \langle g \rangle$ (finite cyclic) in public, then

1. P_i sends $h_i := g^{a_i}$, while hiding $a_i \in \mathbb{Z}$
2. 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**

- Can be converted to PKE [ElGamal 1985]

Is DH Key Exchange Secure?

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

Is DH Key Exchange Secure?

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

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

(DL) Given g, h, find x with $h = g^x$ in G

- Remark: (In)sufficiency is still open

Choice of the Group for Security (1/2)

Choice of the Group for Security (1/2)

Q1. $x \cdot 7 = 15$ in $\mathbb{Z}/16\mathbb{Z}$? ...

Choice of the Group for Security (1/2)

Q1. $x \cdot 7 = 15$ in $\mathbb{Z}/16\mathbb{Z}$? ... $x = 9$

Choice of the Group for Security (1/2)

Q1. $x \cdot 7 = 15$ in $\mathbb{Z}/16\mathbb{Z}$? ... $x = 9$

Q2. $10^x = 6$ in $\mathbb{F}_{17}^{\times}$? ...

Choice of the Group for Security (1/2)

Q1. $x \cdot 7 = 15$ in $\mathbb{Z}/16\mathbb{Z}$? ... $x = 9$

Q2. $10^x = 6$ in $\mathbb{F}_{17}^{\times}$? ... $x = 5$

Choice of the Group for Security (1/2)

Q1. $x \cdot 7 = 15$ in $\mathbb{Z}/16\mathbb{Z}$? ... $x = 9$

Q2. $10^x = 6$ in $\mathbb{F}_{17}^{\times}$? ... $x = 5$

Q2 looks more difficult than Q1,

though $\mathbb{Z}/16\mathbb{Z} \simeq \mathbb{F}_{17}^{\times}$ as groups

Choice of the Group for Security (1/2)

Q1. $x \cdot 7 = 15$ in $\mathbb{Z}/16\mathbb{Z}$? ... $x = 9$

Q2. $10^x = 6$ in $\mathbb{F}_{17}^{\times}$? ... $x = 5$

Q2 looks more difficult than Q1,
though $\mathbb{Z}/16\mathbb{Z} \simeq \mathbb{F}_{17}^{\times}$ as groups

$\Rightarrow$ Difficulty of DL **does depend** on
a "realization" of the same abstract group G!

Choice of the Group for Security (2/2)

Efficient solution for Q1:
use (Extended) Euclidean Algorithm

Choice of the Group for Security (2/2)

Efficient solution for Q1:
use (Extended) Euclidean Algorithm

- which uses integer **division** (or **ordering**)

Choice of the Group for Security (2/2)

Efficient solution for Q1:
use (Extended) Euclidean Algorithm

- which uses integer **division** (or **ordering**)
- for the DL in **additive group** $\mathbb{Z}/n\mathbb{Z}$!

Choice of the Group for Security (2/2)

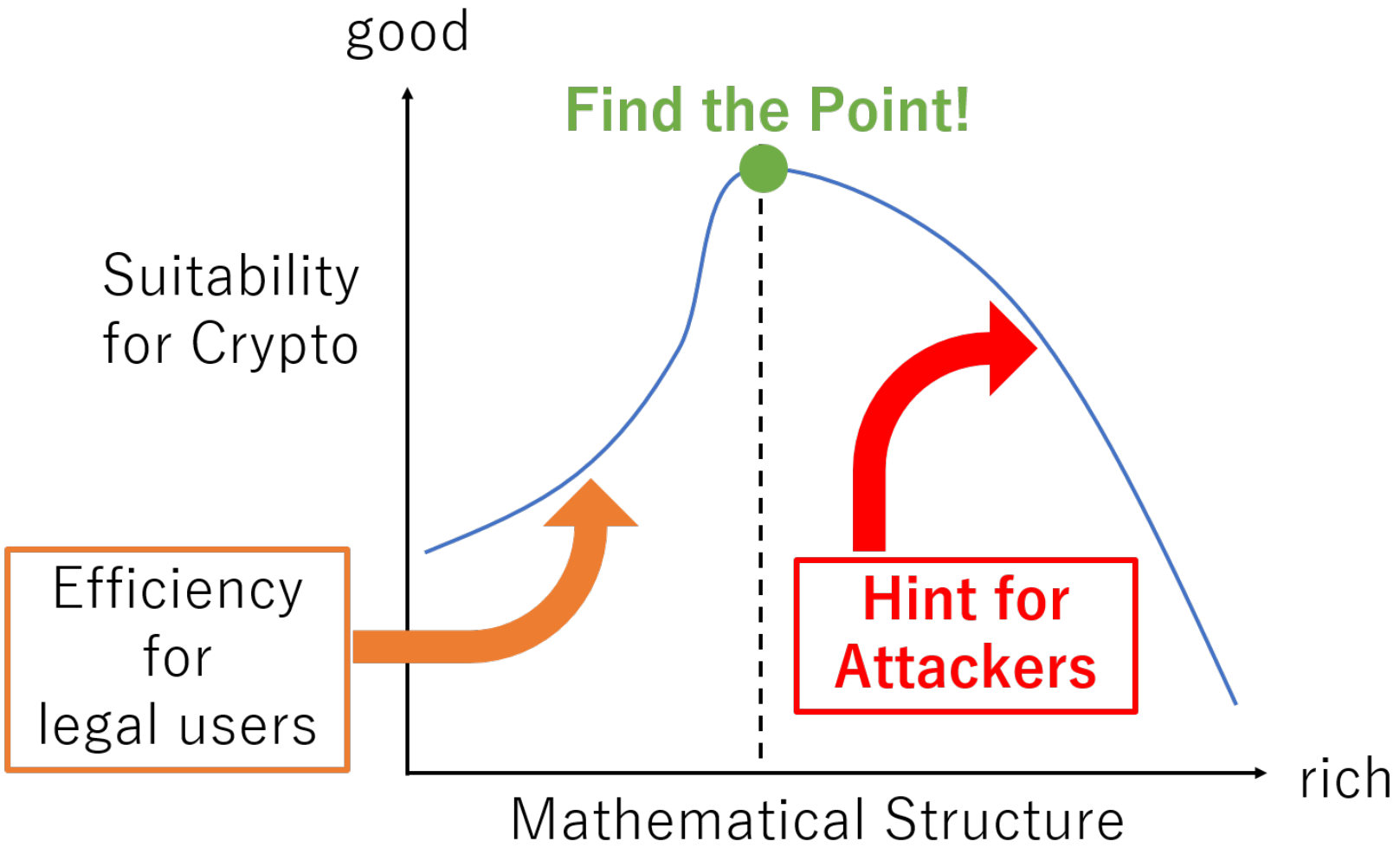
Efficient solution for Q1:
use (Extended) Euclidean Algorithm

- which uses integer **division** (or **ordering**)
- for the DL in **additive group** $\mathbb{Z}/n\mathbb{Z}$!

A lesson: Additional structure for group G makes the DL easier ($\Rightarrow$ break of DH Key Exchange)

- Cf. [Maurer 2005] DL is hard in "generic group"
  - "Oracle access to multiplication table only"

A New Viewpoint from Cryptography



(Mathematician: more structures, more happiness)

State-of-the-Art Candidate from Elliptic Curves

(Subgroups of) groups of rational points on **elliptic curves** (over finite fields)

State-of-the-Art Candidate from Elliptic Curves

(Subgroups of) groups of rational points on **elliptic curves** (over finite fields)

- Additive group structure

State-of-the-Art Candidate from Elliptic Curves

(Subgroups of) groups of rational points on **elliptic curves** (over finite fields)

- Additive group structure
- **Other structures are not known well** (in comparison to $\mathbb{Z}/n\mathbb{Z}$ and ${\mathbb{F}_q}^{\times}$)

State-of-the-Art Candidate from Elliptic Curves

(Subgroups of) groups of rational points on **elliptic curves** (over finite fields)

- Additive group structure
- **Other structures are not known well** (in comparison to $\mathbb{Z}/n\mathbb{Z}$ and ${\mathbb{F}_q}^{\times}$)

Current status: $|G| \gtrapprox 2^{160}$

- Cf. $N \gtrapprox 2^{1024}$ for the RSA

Quantum computer: framework of fast computation using superimposed quantum states

- Not practically implemented so far

Shor's Quantum Algorithms

Quantum computer: framework of fast computation using superimposed quantum states

- Not practically implemented so far

[Shor 1994]: Quantum algorithms, implying

Shor's Quantum Algorithms

Quantum computer: framework of fast computation using superimposed quantum states

- Not practically implemented so far

[Shor 1994]: Quantum algorithms, implying

- integer factoring **in polynomial time**!
- discrete logarithm **in polynomial time**!

(Cf. [Grover 1996]: Search with quadratic speedup)

Shor's main applications: integer factoring and DL

Pinpoint Effect by Shor to Cryptography

Shor's main applications: integer factoring and DL

Main tools of PKE: integer factoring and DL

- Oh, My God!

Pinpoint Effect by Shor to Cryptography

Shor's main applications: integer factoring and DL

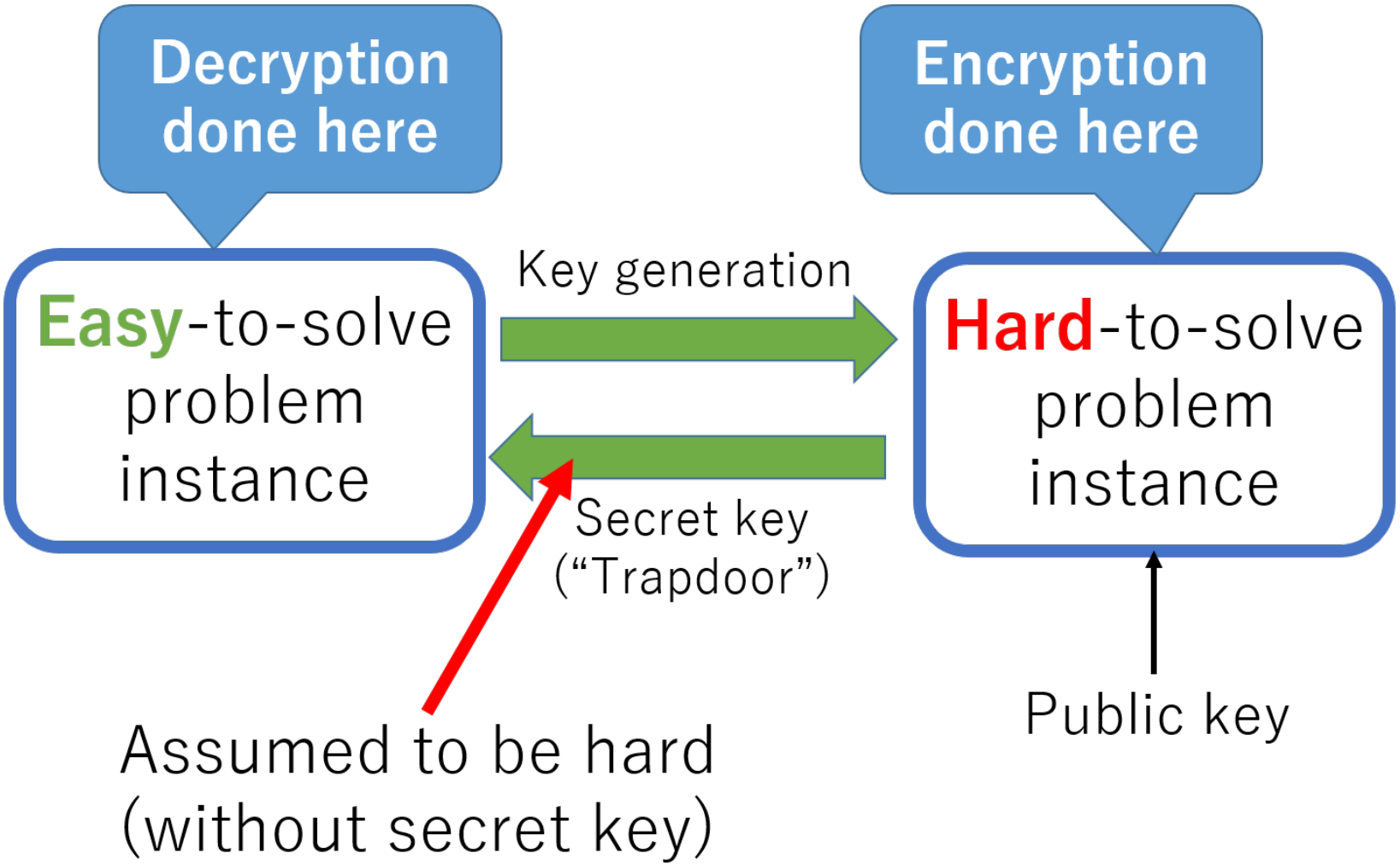
Main tools of PKE: integer factoring and DL

- Oh, My God!

$\longrightarrow$ Importance of "quantum-resistant" PKE

- (Believed to be) unbroken by quantum computer
- Related to the talks by K. Akiyama and K. Takashima

A Major Strategy for PKE



Multi-Party Computation (MPC): Two-Party Case

Given function $f(x, y)$ (e.g., $f(x, y) = \delta_{x,y}$),

- Party P_1 has secret input a_1
- Party P_2 has secret input a_2
- They want to know $f(a_1, a_2)$ by communication

Multi-Party Computation (MPC): Two-Party Case

Given function $f(x, y)$ (e.g., $f(x, y) = \delta_{x,y}$),

- Party P_1 has secret input a_1
- Party P_2 has secret input a_2
- They want to know $f(a_1, a_2)$ by communication
- while **hiding information on each input!**
  - (except those trivially implied from $f(a_1, a_2)$)

A Tool for MPC: Homomorphic Encryption (HE)

Example: additively-HE

A Tool for MPC: Homomorphic Encryption (HE)

Example: additively-HE

- Message set $\mathcal{M}$ is additive group

A Tool for MPC: Homomorphic Encryption (HE)

Example: additively-HE

- Message set $\mathcal{M}$ is additive group
- A "practical" operation $\boxplus$ for ciphertexts with

$$\mathrm{Dec}(c_1 \boxplus c_2) = \mathrm{Dec}(c_1) + \mathrm{Dec}(c_2) \in \mathcal{M}$$

(called "homomorphic operation")

A Tool for MPC: Homomorphic Encryption (HE)

Example: additively-HE

- Message set $\mathcal{M}$ is additive group
- A "practical" operation $\boxplus$ for ciphertexts with

$$\mathrm{Dec}(c_1 \boxplus c_2) = \mathrm{Dec}(c_1) + \mathrm{Dec}(c_2) \in \mathcal{M}$$

  (called "homomorphic operation")
- "Messages can be added in encrypted form"

A Tool for MPC: Homomorphic Encryption (HE)

Example: additively-HE

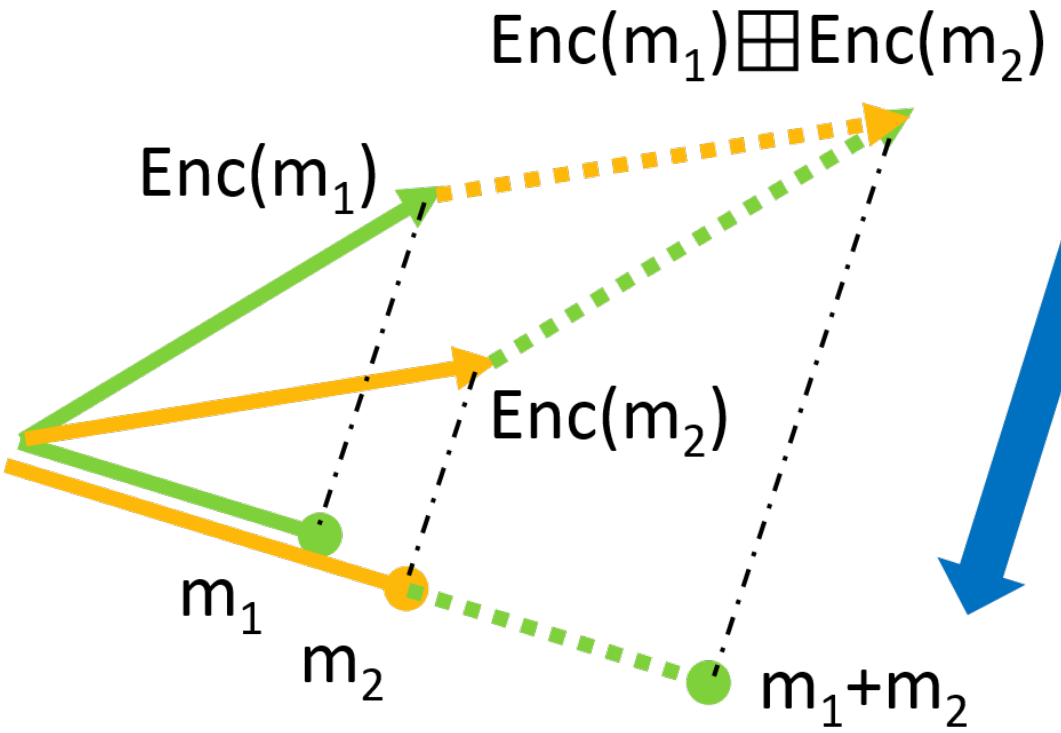
- Message set $\mathcal{M}$ is additive group
- A "practical" operation $\boxplus$ for ciphertexts with

$$\mathrm{Dec}(c_1 \boxplus c_2) = \mathrm{Dec}(c_1) + \mathrm{Dec}(c_2) \in \mathcal{M}$$

  (called "homomorphic operation")
- "Messages can be added in encrypted form"

Related to the talks by K. Shimizu, H. Arimura (and K. Akiyama)

A "Rough Idea" for HE



Direction should be hard to find without key (art of cryptographer)

Projection in **this direction** yields decryption

Example of (Multiplicative) HE [ElGamal 1985]

Example of (Multiplicative) HE [ElGamal 1985]

Public key: $G = \langle g \rangle$ (prime order), $h \in G$
Secret key: $s \in \mathbb{Z}$ with $h = g^s$

Example of (Multiplicative) HE [ElGamal 1985]

Public key: $G = \langle g \rangle$ (prime order), $h \in G$

Secret key: $s \in \mathbb{Z}$ with $h = g^s$

- Given $m \in G$, $\mathrm{Enc}(m) := (g^r, h^r m) \in G^2$
  - where $r \in \mathbb{Z}$ is random

Example of (Multiplicative) HE [ElGamal 1985]

Public key: $G = \langle g \rangle$ (prime order), $h \in G$
Secret key: $s \in \mathbb{Z}$ with $h = g^s$

- Given $m \in G$, $\mathrm{Enc}(m) := (g^r, h^r m) \in G^2$
  - where $r \in \mathbb{Z}$ is random
- Given $c = (c_1, c_2)$, $\mathrm{Dec}(c) := {c_1}^{-s} c_2$
  - "Project to $(g^0, g^{\mathbb{Z}})$ in direction (g^1, g^{-s})"

Example of (Multiplicative) HE [ElGamal 1985]

Public key: $G = \langle g \rangle$ (prime order), $h \in G$
Secret key: $s \in \mathbb{Z}$ with $h = g^s$

- Given $m \in G$, $\mathrm{Enc}(m) := (g^r, h^r m) \in G^2$
  - where $r \in \mathbb{Z}$ is random
- Given $c = (c_1, c_2)$, $\mathrm{Dec}(c) := {c_1}^{-s} c_2$
  - "Project to $(g^0, g^{\mathbb{Z}})$ in direction (g^1, g^{-s})"
- Homomorphic operation: multiplication in G^2

Example of MPC from HE

How to compute δ_{a_1,a_2} (Notation: $[[a]] := \mathrm{Enc}(a)$)

Suppose: additively-HE with $\mathcal{M} = \mathbb{F}_p$

Example of MPC from HE

How to compute δ_{a_1,a_2} (Notation: $[[a]] := \mathrm{Enc}(a)$)

Suppose: additively-HE with $\mathcal{M} = \mathbb{F}_p$

1. P_1 chooses key, sends public key only

Example of MPC from HE

How to compute δ_{a_1,a_2} (Notation: $[[a]] := \mathrm{Enc}(a)$)

Suppose: additively-HE with $\mathcal{M} = \mathbb{F}_p$

1. P_1 chooses key, sends public key only
2. P_1 generates and sends $[[a_1]]$

Example of MPC from HE

How to compute δ_{a_1,a_2} (Notation: $[[a]] := \mathrm{Enc}(a)$)

Suppose: additively-HE with $\mathcal{M} = \mathbb{F}_p$

1. P_1 chooses key, sends public key only
2. P_1 generates and sends $[[a_1]]$
3. P_2 computes $[[a_1]] \boxplus [[-a_2]] = [[a_1 - a_2]]$

Example of MPC from HE

How to compute δ_{a_1,a_2} (Notation: $[[a]] := \mathrm{Enc}(a)$)

Suppose: additively-HE with $\mathcal{M} = \mathbb{F}_p$

1. P_1 chooses key, sends public key only
2. P_1 generates and sends $[[a_1]]$
3. P_2 computes $[[a_1]] \boxplus [[-a_2]] = [[a_1 - a_2]]$
4. P_2 computes $[[r(a_1 - a_2)]]$ for random $r \neq 0$
   - by random iteration of $\boxplus$ to $[[a_1 - a_2]]$

Example of MPC from HE

How to compute δ_{a_1,a_2} (Notation: $[[a]] := \mathrm{Enc}(a)$)

Suppose: additively-HE with $\mathcal{M} = \mathbb{F}_p$

1. P_1 chooses key, sends public key only
2. P_1 generates and sends $[[a_1]]$
3. P_2 computes $[[a_1]] \boxplus [[-a_2]] = [[a_1 - a_2]]$
4. P_2 computes $[[r(a_1 - a_2)]]$ for random $r \neq 0$
   - by random iteration of $\boxplus$ to $[[a_1 - a_2]]$
5. P_1 decrypts $[[r(a_1 - a_2)]] \rightsquigarrow 0$ iff $a_1 = a_2$

Example of MPC from HE

How to compute δ_{a_1,a_2} (Notation: $[[a]] := \mathrm{Enc}(a)$)

Suppose: additively-HE with $\mathcal{M} = \mathbb{F}_p$

1. P_1 chooses key, sends public key only
2. P_1 generates and sends $[[a_1]]$
3. P_2 computes $[[a_1]] \boxplus [[-a_2]] = [[a_1 - a_2]]$
4. P_2 computes $[[r(a_1 - a_2)]]$ for random $r \neq 0$
   - by random iteration of $\boxplus$ to $[[a_1 - a_2]]$
5. P_1 decrypts $[[r(a_1 - a_2)]] \rightsquigarrow 0$ iff $a_1 = a_2$

Applications to bioinformatics: Talk by K. Shimizu

Fully Homomorphic Encryption [2009]

(Additively-)HE: “addition in encrypted form”

Fully Homomorphic Encryption [2009]

(Additively-)HE: "addition in encrypted form"

Fully homomorphic encryption (FHE):

Any computation in encrypted form

Fully Homomorphic Encryption [2009]

(Additively-)HE: "addition in encrypted form"

Fully homomorphic encryption (FHE):

Any computation in encrypted form

- $\Leftrightarrow$ Ring-HE, when $\mathcal{M} = \mathbb{F}_p$ (p prime)
  - $\rightarrow$ Use of FHE is related to polynomial expressions of functions (talk by T. Maeno)

(Too) Simplified Example [2010] [N. et al. 2015]

$\mathbb{Z}/\ell\mathbb{Z}$ identified with $\{0, \ldots, \ell - 1\}$ by "mod"

Choose $p' \gg p$ primes, $p' \mid N$

(Too) Simplified Example [2010] [N. et al. 2015]

$\mathbb{Z}/\ell\mathbb{Z}$ identified with $\{0, \ldots, \ell - 1\}$ by "mod"

Choose $p' \gg p$ primes, $p' \mid N$

$\mathrm{Enc}(m) = r'p' + rp + m \bmod N$ for $m \in \mathbb{F}_p$

(Too) Simplified Example [2010] [N. et al. 2015]

$\mathbb{Z}/\ell\mathbb{Z}$ identified with $\{0, \ldots, \ell - 1\}$ by "mod"

Choose $p' \gg p$ primes, $p' \mid N$

$\mathrm{Enc}(m) = r'p' + rp + m \bmod N$ for $m \in \mathbb{F}_p$

$\mathrm{Dec}(c) = (c \bmod p') \bmod p$

- Decryption works iff r is "not too large"

(Too) Simplified Example [2010] [N. et al. 2015]

$\mathbb{Z}/\ell\mathbb{Z}$ identified with $\{0, \ldots, \ell - 1\}$ by "mod"

Choose $p' \gg p$ primes, $p' \mid N$

$\mathrm{Enc}(m) = r'p' + rp + m \bmod N$ for $m \in \mathbb{F}_p$

$\mathrm{Dec}(c) = (c \bmod p') \bmod p$

- Decryption works iff r is "not too large"

Ring-homomorphic operations: as usual in $\mathbb{Z}/N\mathbb{Z}$

- but iteration of operations is limited! (r grows)

(Too) Simplified Example [2010] [N. et al. 2015]

$\mathbb{Z}/\ell\mathbb{Z}$ identified with $\{0, \ldots, \ell - 1\}$ by "mod"

Choose $p' \gg p$ primes, $p' \mid N$

$\mathrm{Enc}(m) = r'p' + rp + m \bmod N$ for $m \in \mathbb{F}_p$

$\mathrm{Dec}(c) = (c \bmod p') \bmod p$

- Decryption works iff r is "not too large"

Ring-homomorphic operations: as usual in $\mathbb{Z}/N\mathbb{Z}$

- but iteration of operations is limited! (r grows)

"Bootstrapping": refreshing the ciphertext

- possible, but very inefficient

Towards "Unlimited" Ring-HE

A (hopefully) possible strategy:

Towards "Unlimited" Ring-HE

A (hopefully) possible strategy:

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

Towards "Unlimited" Ring-HE

A (hopefully) possible strategy:

1. "Embed" $\mathbb{F}_p$ into a (non-commutative) group G
   - Operations of $\mathbb{F}_p$ realized by operations of G
2. Take a lift of G (e.g., $G \times H$ for suitable H)

Towards "Unlimited" Ring-HE

A (hopefully) possible strategy:

1. "Embed" $\mathbb{F}_p$ into a (non-commutative) group G
   - Operations of $\mathbb{F}_p$ realized by operations of G
2. Take a lift of G (e.g., $G \times H$ for suitable H)
3. "Homomorphically hide" the structure of the lift

Towards "Unlimited" Ring-HE

A (hopefully) possible strategy:

1. "Embed" $\mathbb{F}_p$ into a (non-commutative) group G
   - Operations of $\mathbb{F}_p$ realized by operations of G
2. Take a lift of G (e.g., $G \times H$ for suitable H)
3. "Homomorphically hide" the structure of the lift

$\rightsquigarrow$ hard-to-compute group hom. $\varphi\colon \widetilde{G} \twoheadrightarrow G$

- must be easy-to-compute with secret key
- Public: G and generators of $\ker \varphi$ (for Enc)

How to Realize NAND Gate in Simple Groups

$$\mathrm{NAND}(b_1, b_2) = 0 \text{ iff } b_1 = b_2 = 1$$

How to Realize NAND Gate in Simple Groups

$\mathrm{NAND}(b_1, b_2) = 0$ iff $b_1 = b_2 = 1$

[Ostrovsky–Skeith 2008] For any non-commutative finite simple group G, there exist $g_0 \neq g_1 \in G$ and $F\colon G^2 \to G$ with:

How to Realize NAND Gate in Simple Groups

$\mathrm{NAND}(b_1, b_2) = 0$ iff $b_1 = b_2 = 1$

[Ostrovsky–Skeith 2008] For any non-commutative finite simple group G, there exist $g_0 \neq g_1 \in G$ and $F\colon G^2 \to G$ with:

- $F(g_1, g_1) = g_0$
- $F(g_0, g_0) = F(g_0, g_1) = F(g_1, g_0) = g_1$

How to Realize NAND Gate in Simple Groups

$\mathrm{NAND}(b_1, b_2) = 0$ iff $b_1 = b_2 = 1$

[Ostrovsky–Skeith 2008] For any non-commutative finite simple group G, there exist $g_0 \neq g_1 \in G$ and $F\colon G^2 \to G$ with:

- $F(g_1, g_1) = g_0$
- $F(g_0, g_0) = F(g_0, g_1) = F(g_1, g_0) = g_1$
- **F is composed of group operations in G**

How to Realize NAND Gate in Simple Groups

$\mathrm{NAND}(b_1, b_2) = 0$ iff $b_1 = b_2 = 1$

[Ostrovsky–Skeith 2008] For any non-commutative finite simple group G, there exist $g_0 \neq g_1 \in G$ and $F\colon G^2 \to G$ with:

- $F(g_1, g_1) = g_0$
- $F(g_0, g_0) = F(g_0, g_1) = F(g_1, g_0) = g_1$
- **F is composed of group operations in G**

Proof (sketch):

- $\langle\{[*, \sigma_1]\}\rangle$ is normal, hence $= G$
- $\Rightarrow \exists(\text{compositions of } [*, \sigma_1]^{\pm 1}) = \sigma_1$
- When $\sigma_1 \mapsto \sigma_0$, LHS becomes $1 = \sigma_0$ $\square$

Towards Homomorphically Hiding the Group

My recent (very rough) idea:

Towards Homomorphically Hiding the Group

My recent (very rough) idea:

1. Presentation of $G \times H$ by generators/relations

Towards Homomorphically Hiding the Group

My recent (very rough) idea:

1. Presentation of $G \times H$ by generators/relations
2. "Shuffle" presentation by randomly applying Tietze transformations

Towards Homomorphically Hiding the Group

My recent (very rough) idea:

1. Presentation of $G \times H$ by generators/relations
2. "Shuffle" presentation by randomly applying Tietze transformations
3. Apply Knuth-Bendix completion algorithm, to yield normal form of each group element
    - Otherwise, Enc is also hard-to-compute

Towards Homomorphically Hiding the Group

My recent (very rough) idea:

1. Presentation of $G \times H$ by generators/relations
2. "Shuffle" presentation by randomly applying Tietze transformations
3. Apply Knuth-Bendix completion algorithm, to yield normal form of each group element
    - Otherwise, Enc is also hard-to-compute

Current problems:

- Knuth-Bendix algorithm may not terminate
- Is it really secure?

Other Topics (1) Set Families and Cryptosystems

- t-cover-free family: Each set is not covered by union of any other t sets
- Application to "stronger" encryption: (Roughly speaking) ciphertext is secure even if other t ciphertexts are decoded by attacker
  - Idea: A set indicates "set of encryption keys for a ciphertext", then at least one key remains safe even when t ciphertexts are stolen

Other Topics (2) Polynomial Interpolation and Secret Sharing

- Fact: Degree-n polynomial is uniquely determined by $n+1$ points
- Applicable to secret sharing [Shamir 1979]
  - Each user holds one point (at outside y-axis)
  - Sufficient # of users can recover the polynomial, whose y-intercept is the secret value
  - Insufficient # of points have no info. on the polynomial (y-intercept can be still arbitrary), hence secret
- Related to the talk by Y. Suga

Other Topics (3) Homomorphic Encryption from Ideal Class Groups

- Additive HE based on ideal class groups of (non-maximal order of) imaginary quadratic fields $\mathbb{Q}(\sqrt{D})$ (in CT-RSA 2015)
- Security under consideration: Efficient algorithm to (approximately) compute the class number will break the cryptosystem

Other Topics (4) Gaussian Quadrature and Content Protection

- Gaussian quadrature: Express integral of polynomial over interval as a finite weighted sum of polynomial values (at some specific points)
- Such choice of points/weights can be applied to good parameter choice in some cryptographic scheme for content protection (digital rights management) [N. et al. 2007]

Other Topics (5) abc Conjecture for Cryptographic Study

- Recent study by Y. Hashimoto, K. Shinagawa, N. et al.: Some kind of cryptographic algorithm by physical cards
- A part of result: A lower bound for such algorithms in certain setting, derived from abc Conjecture, Prime Number Theorem, König's Lemma, etc.
- Affimative result also: Application of "permutations of same type are conjugate to each other"
- To be presented in SCIS 2017

Conclusion

Q. What kind of math. are useful in crypto. (and/or other "practical" topics)?

A. Unpredictable!

- I am now trying to apply techniques from set theory, computability theory, ...