ISSN 2188-286X

マス・フォア・インダストリ研究　No.9

代数的手法による数理暗号解析
Workshop on analysis of mathematical cryptography via algebraic methods

Institute of Mathematics for Industry
Kyushu University

編　集　阿部　拓郎
高島　克幸
縫田　光司
安田　雅哉

九州大学マス・フォア・インダストリ研究所

About the Mathematics for Industry Research

The Mathematics for Industry Research was founded on the occasion of the certification of the Institute of Mathematics for Industry (IMI), established in April 2011, as a MEXT Joint Usage/Research Center – the Joint Research Center for Advanced and Fundamental Mathematics for Industry – by the Ministry of Education, Culture, Sports, Science and Technology (MEXT) in April 2013. This series publishes mainly proceedings of workshops and conferences on Mathematics for Industry (MfI). Each volume includes surveys and reviews of MfI from new viewpoints as well as up-to-date research studies to support the development of MfI.

October 2014
Yasuhide Fukumoto
Director
Institute of Mathematics for Industry

Workshop on analysis of mathematical cryptography via algebraic methods

Mathematics for Industry Research No.9, Institute of Mathematics for Industry, Kyushu University
ISSN 2188-286X
Editors: Takuro Abe, Katsuyuki Takashima, Koji Nuida, Masaya Yasuda
Date of issue: 1 March 2018
Publisher:
Institute of Mathematics for Industry, Kyushu University
Motooka 744, Nishi-ku, Fukuoka, 819-0395, JAPAN
Tel +81-(0)92-802-4402, Fax +81-(0)92-802-4405
URL http://www.imi.kyushu-u.ac.jp/
Printed by
Social Welfare Service Corporation Fukuoka Colony
1-11-1, Midorigahama, Shingu-machi Kasuya-gun, Fukuoka, 811-0119, Japan
TEL +81-(0)92-962-0764 FAX +81-(0)92-962-0768

代数的手法による数理暗号解析

Workshop on analysis of mathematical cryptography via algebraic methods

編集

阿部　拓郎
高島　克幸
縫田　光司
安田　雅哉

平成 30 年 2 月 19 日作成

巻頭言

高度かつ急速に進化する現代情報社会において，情報セキュリティを支える暗号技術は必須であり，その安全性はある数学問題の計算量困難性に依存している．例えば，現在普及している RSA 暗号と楕円曲線暗号の安全性は，それぞれ素因数分解問題と楕円曲線暗号離散対数問題の計算量困難性に基づいている．一方，現在研究開発が活発に行われている量子計算機が実現すると，これらの暗号技術が危殆化することが知られており，量子計算機でも耐性を持つ「ポスト量子暗号」の研究が盛んに行われている．実際、2016 年 2 月に開催された国際会議 PQCrypto2016 において，米国標準技術研究所 NIST はポスト量子暗号の標準化計画を発表し，2017 年 11 月末に公募〆切を迎え，格子暗号・符号ベース暗号・多変数公開鍵暗号などの数学ベースの暗号方式が多数提案された．このように暗号と数学は密接に関係する分野であり，両分野の研究交流・議論が常に必要である．

本研究集会では，暗号と代数学の研究者間の積極的な交流を促すことを目的とする．より具体的には，双方の各専門分野における高度な専門知識・最新情報を共有すると共に，互いの研究課題に対して他分野からのアプローチを試みることで．既存研究では得られなかった新しい研究の芽や方向性を探索することを目的とする．本研究集会において，暗号と代数学の研究者からの講演内容（全 11 件）は以下である：

- 暗号研究者からの講演内容：格子暗号方式に対するサイドチャンネル攻撃，代数体の整数環上の格子基底簡約，NIST 公募提案の符号ベース暗号方式の紹介，LLL 基底簡約を用いた RSA 暗号解読，群論からの完全準同型暗号の構成アプローチ，多変数公開鍵暗号とその攻撃の紹介，分解体上の Ring-LWE 問題ベースの準同型暗号に対する格子攻撃評価
- 代数学研究者からの講演内容：Fine-grained による計算量と暗号との関係，monomial GAPN 関数とその分類理論，虚数乗法を持つ楕円曲線上の素因数分解アプローチ，離散フーリエ変換と符号

今回の講演では，格子暗号・符号暗号・多変数公開鍵暗号などの NIST 公募に提案されているポスト量子暗号に関する提案方式や攻撃可能性などの講演があり，暗号と代数学の両分野からの研究アプローチ・考え方の違いを積極的に議論することができた．また数学の暗号応用として，離散フーリエ変換や GAPN 関数による符号への応用や，群論を用いた完全準同型暗号の構成などの新しい研究の芽を共有でき，今後の研究の方向性・可能性を見つけることができた．また本研究集会では，産官学のそれぞれで著名な研究者に数多く参加して頂け，暗号と数学の両分野における長期的かつ永続的な研究交流を促進するための研究集会とすることができた．今後も暗号と数学の交流研究集会を定期的に開催することで，研究分野間のシナジー効果を促すと共に，産官学間の長期的な連携協力の礎にしていければ幸いである．

世話人
阿部　拓郎（九州大学マス・フォア・インダストリ研究所）
高島　克幸（三菱電機株式会社 情報技術総合研究所）
縫田　光司（産業技術総合研究所 / JST さきがけ）
安田　雅哉（九州大学マス・フォア・インダストリ研究所）

九州大学 IMI 共同利用・研究集会（II）

代数的手法による数理暗号解析
Workshop on analysis of mathematical cryptography via algebraic methods

Date : February 5 (Mon) - February 7 (Wed), 2018
Venue : Meeting RoomA Nishijin Plaza, Kyushu University
2-16-23, Nishijin, Sawara-ku, Fukuoka-shi, Fukuoka, 814-0002 JAPAN
URL : http://www.imi.kyushu-u.ac.jp/events/view/2227

February 5 (Mon)

13:00	**Reception**
13:15 - 13:25	**Opening Remarks**

13:30 - 14:30 **Speaker : Mehdi Tibouchi (NTT)**

Physical attacks on lattice-based schemes

Abstract:

As the NIST competition on postquantum cryptography begins, it becomes increasingly important to understand not just the theoretical, black-box security of lattice-based schemes, but also the security of implementations. In this talk, we will discuss recent developments in this area, and particularly fault and side-channel attacks on lattice-based signatures, some of which involved interesting mathematical techniques.

14:45 - 15:45 **Speaker : Kim Taechan (NTT)**

Use of algebraic subfield structure in cryptanalysis

Abstract:

In this talk, we explain how the algebraic subfield structure can be exploited to obtain more efficient cryptanalysis in many cryptosystems. Firstly, we describe ”extended tower number field sieve” method (based on my work at Crypto2016 and PKC2017) that leads a significant security loss in pairing-based cryptosystems using subfield structures of finite fields. In addition, we also present that lattice reduction algorithms (e.g. LLL algorithm) can be accelerated when the lattices are defined over a number field that contains a certain subfield (whose ring of integers are Euclidean ring). The later topic is based on my recent work that appeared at IMA conference on Cryptography and Coding.

16:00 - 17:00 **Speaker : Suguru Tamaki (Kyoto University)**

Fine-grained complexity and cryptography: A personal survey

Abstract:

A major goal of computational complexity theory is to classify computational problems into tractable and intractable ones. The most adopted definition of tractability is polynomial time solvability.

Problems are shown to be intractable based on assumptions such as "P is not equal to NP" or "integer factorization requires super-polynomial time".

The goal of fine-grained complexity theory is to determine more precise complexities of computational problems using more quantitative but plausible hardness assumptions. Recently we have seen lots of exciting algorithmic and hardness results in this rapidly developing field. I will present a personal survey on fine-grained complexity focusing on topics related to cryptography such as the shortest and closest vector problems, systems of multivariate polynomial equations and fine-grained average-case hardness.

February 6 (Tue)

9:00 **Reception**

9:30 - 10:30 **Speaker : Masamichi Kuroda (Hokkaido University)**

On monomial GAPN (Generalized Almost Perfect Nonlinear) functions and their classification

Abstract:

APN (Almost Perfect Nonlinear) functions on finite fields of characteristic two have useful properties and applications in cryptography, coding theory, finite geometry and so on. On the other hand, APN functions for odd characteristic have quite different algebraic properties. GAPN (Generalized APN) functions were defined to satisfy some generalizations of basic properties of APN functions for even characteristic [K and Tsujie, FFA vol. 47, 2017]. In this talk, we will introduce monomial GAPN functions and their partial classification. This study is based on a joint work with Shuhei Tsujie (Hokkaido University).

10:40 - 11:40 **Speaker : Yusuke Aikawa (Hokkaido University)**

Elliptic curve method with complex multiplication method

Abstract:

In SCIS 2017, M. Shirase proposed a new factoring algorithm for integers by combining elliptic curve method (ECM) with complex multiplication method which is one of generating methods of elliptic curves. This algorithm works in polynomial time for a composite having a prime factor of special form which is related to the complex multiplication theory. However, the range of application of this algorithm is limited. We give a generalization and extend the range of application. In this talk, firstly I will give a brief explanation of ECM and complex multiplication theory. After that, I will explain the generalized algorithm. This is a joint work with K. Nuida and M. Shirase.

13:10 - 14:10 **Speaker : Norihiro Nakashima (Tokyo Denki University)**

A modification of the discrete Fourier transform for the code defined by Garcia-Stichtenoth tower

Abstract:

A decoding algorithm for algebraic geometry codes is proposed, using the discrete Fourier transform and Berlekamp-Massey-Sakata algorithm. Meanwhile Garcia and Stichtenoth explicitly constructed a tower of algebraic curves which attains the upper bound of Drinfeld-Vladut bound. In this talk, I present a method to reduce the computational complexity of the discrete Fourier transform for the algebraic geometry codes defined by Garcia-Stichtenoth tower. A key of this reduction is to give affine rational points for Garcia-Stichtenoth tower.
This is a joint work with H. Matsui.

14:20 - 15:20 **Speaker : Carlos Cid (Royal Holloway, University of London)**

Code-based cryptography: design and security

Abstract:

In 1978, Robert McEliece proposed a public-key encryption scheme based on error-correcting codes. The McEliece scheme (and its variant due to Niederreiter) is a simple, elegant and efficient design, and has its security based on two hardness assumptions: the intractability of decoding a random linear code, and the difficulty of distinguishing some permuted linear binary codes from a random code. McEliece’s construction is over 40 years, and despite enormous cumulative efforts by the cryptographic community, it remains unbroken when instantiated with Goppa codes for suitable parameters. Its main drawback is the large public key, and attempts to reduce it to more manageable sizes have often resulted on insecure designs. Code-based cryptography is again attracting considerable attention from the Cryptographic community, mainly due to the ongoing NIST PQ competition: over 20 submissions are based on errorcorrecting codes. In this talk we give an overview of code-based cryptography, main designs and their security, and discuss a selected few submissions to the NIST competition.

15:30 - 16:30 **Speaker : Atsushi Takayasu (The University of Tokyo)**

Solving RSA and factoring problems using LLL reduction

Abstract:

In 1996, Coppersmith introduced lattice-based methods for finding small roots of modular polynomials. By using the method, a number of vulnerability of RSA have been reported so far. In this talk, I explain the basic approach of the method. Then, I introduce our attack on small CRT-exponent RSA. The attacks improve previous ones proposed by Bleichenbacher-May (PKC'06) and Jochemsz-May (Crypto'07). In general, to recover as large root as possible, we should design appropriate lattices that relate to algebraic structures of the target polynomials. We obtain the results by exploiting additional algebraic structures in a clever way.

16:40 - 17:40 **Speaker : Koji Nuida (AIST/JST PRESTO)**

Towards fully homomorphic encryption without ciphertext noise from group theory

Abstract:

Fully homomorphic encryption (FHE) is a kind of (public key) encryption scheme that allows anyone to perform arbitrary operations on plain-texts via certain special operations on the corresponding ciphertexts. In 2008, Ostrovsky and Skeith III suggested an approach towards achieving FHE from group-theoretic viewpoint, but no observations on how to actually construct FHE based on their approach have been given so far. In this talk, I explain my recent work based on this approach, which is still incomplete but would show several potential, interesting connections between group theory and cryptography.

18:10 **Banquet**

February 7 (Wed)

9:00 **Reception**

9:30 - 10:30 **Speaker : Yasufumi Hashimoto (University of Ryukyu)**

A survey on multivariate public key cryptosystem

Abstract:

A multivariate public key cryptosystem (MPKC) is a public key cryptosystem whose public key is a set of multivariate quadratic forms over a finite field. It has been considered to be one of candidates of Post Quantum Cryptographies. From 1980s to now, various MPKCs have been proposed and some of them were already broken. In this talk, we give a survey on MPKCs.

10:45 - 11:45 **Speaker : Shinya Okumura (Osaka University)**

On the Security of Homomorphic Encryption Schemes Based on Ring-LWE Problem over Decomposition Fields

Abstract:

Ring-LWE problem has been an important tool in cryptography to construct cryptosystems, key exchange protocols and homomorphic encryption schemes, which are expected to be secure against attacks by quantum computers. Cyclotomic fields are always used as underlying number fields of Ring-LWE problem from the viewpoints of security and efficiency. However, especially, in the case of homomorphic encryption schemes, improving the efficiency is still required. Arita and Handa proposed to use certain subfields of cyclotomic fields with prime conductors, called decomposition fields, as underlying number fields of Ring-LWE problem to construct a homomorphic encryption scheme at ICISC 2017. Their homomorphic encryption scheme can provide many plaintext slots in which homomorphic arithmetics are easily executed. However, Arita et al. did not analyze the security of Ring-LWE problem over decomposition fields. In this talk, we will present experimental results on attacks using lattices and ring structures against Ring-LWE problem over cyclotomic fields (with prime conductors) and decomposition fields, which indicate that Arita et al.'s homomorphic encryption scheme would be as secure as previous ones. This is a joint work with Shota Terada, Hideto Nakano and Atsuko Miyaji (Osaka University).

Organizing Committee

Takuro Abe(Kyushu Univiersity)
Katsuyuki Takashima(Mitsubishi Electric Corporation)
Koji Nuida(AIST/JST PRESTO)
Masaya Yasuda(Kyushu University)

Table of contents

Physical Attacks Against Lattice-Based Schemes

Mehdi Tibouchi

NTT Secure Platform Laboratories

代数的手法による数理暗号解析, 2018–2–5

joint work with T. Espitau, P.-A. Fouque and B. Gérard

Outline

Introduction
- Implementation attacks
- Implementation attacks on lattice schemes

Physical attacks against BLISS
- The BLISS signature scheme
- Fault attack on the Gaussian sampling
- SCA on the rejection sampling

Conclusion and countermeasures

Outline

Introduction

- Implementation attacks
- Implementation attacks on lattice schemes

Physical attacks against BLISS

- The BLISS signature scheme
- Fault attack on the Gaussian sampling
- SCA on the rejection sampling

Conclusion and countermeasures

Breaking provable crypto is harder

- Most crypto proposed in the last 15–20 years: provably secure
- Breaking it = provably as hard as solving some algorithmic problem like integer factorization or computing discrete logs
- Hence, cryptanalysis = major algorithmic advance?

Breaking provable crypto is harder

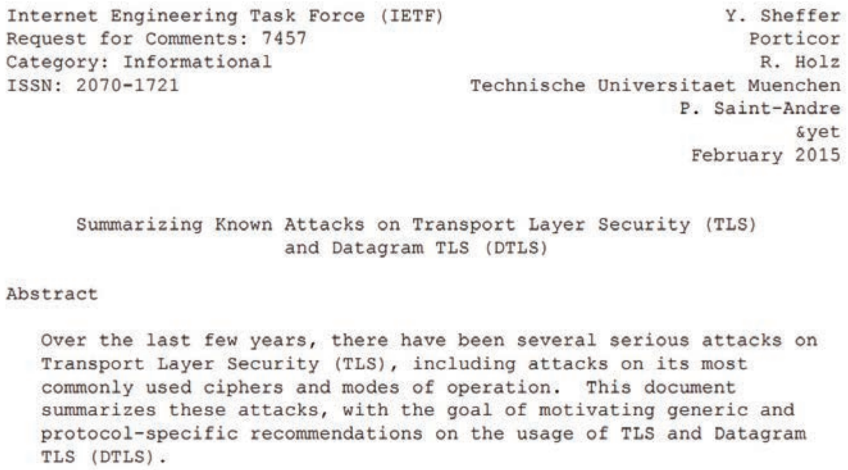
- Most crypto proposed in the last 15–20 years: provably secure
- Breaking it = provably as hard as solving some algorithmic problem like integer factorization or computing discrete logs
- Hence, cryptanalysis = major algorithmic advance?

Breaking provable crypto is harder

- Most crypto proposed in the last 15–20 years: provably secure
- Breaking it = provably as hard as solving some algorithmic problem like integer factorization or computing discrete logs
- Hence, cryptanalysis = major algorithmic advance?

Yet, many attacks against deployed crypto

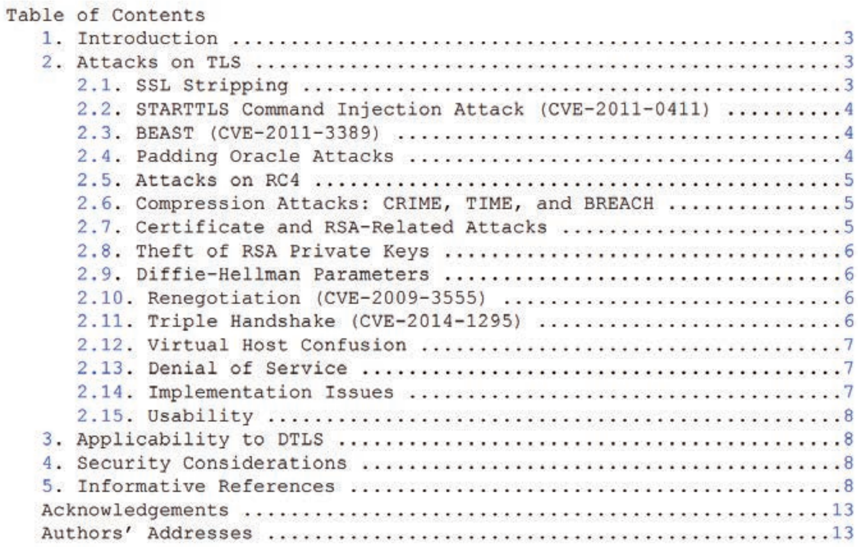
The crypto protocol that is perhaps most used in everyday life, TLS, is attacked all the time!



```
Internet Engineering Task Force (IETF)                        Y. Sheffer
Request for Comments: 7457                                       Porticor
Category: Informational                                          R. Holz
ISSN: 2070-1721                         Technische Universitaet Muenchen
                                                          P. Saint-Andre
                                                                    &yet
                                                           February 2015


     Summarizing Known Attacks on Transport Layer Security (TLS)
                        and Datagram TLS (DTLS)

Abstract

   Over the last few years, there have been several serious attacks on
   Transport Layer Security (TLS), including attacks on its most
   commonly used ciphers and modes of operation.  This document
   summarizes these attacks, with the goal of motivating generic and
   protocol-specific recommendations on the usage of TLS and Datagram
   TLS (DTLS).
```

Yet, many attacks against deployed crypto

The crypto protocol that is perhaps most used in everyday life, TLS, is attacked all the time!



```
Table of Contents
   1. Introduction ....................................................3
   2. Attacks on TLS ..................................................3
      2.1. SSL Stripping ..............................................3
      2.2. STARTTLS Command Injection Attack (CVE-2011-0411) ..........4
      2.3. BEAST (CVE-2011-3389) ......................................4
      2.4. Padding Oracle Attacks .....................................4
      2.5. Attacks on RC4 .............................................5
      2.6. Compression Attacks: CRIME, TIME, and BREACH ...............5
      2.7. Certificate and RSA-Related Attacks ........................5
      2.8. Theft of RSA Private Keys ..................................6
      2.9. Diffie-Hellman Parameters ..................................6
      2.10. Renegotiation (CVE-2009-3555) .............................6
      2.11. Triple Handshake (CVE-2014-1295) ..........................6
      2.12. Virtual Host Confusion ....................................7
      2.13. Denial of Service .........................................7
      2.14. Implementation Issues .....................................7
      2.15. Usability .................................................8
   3. Applicability to DTLS ...........................................8
   4. Security Considerations .........................................8
   5. Informative References ..........................................8
   Acknowledgements ..................................................13
   Authors' Addresses ................................................13
```

So how do people actually break crypto?

- Very rarely: major algorithmic improvement
  - Biggest one recently: progress on small characteristic discrete logarithms/pairings
- More commonly: non-provably secure schemes shown to be insecure
  - Several of the TLS attacks
  - Many legacy scheme still in use could be broken (e.g. PKCS#1v1.5 signatures?)
- Most importantly: implementation attacks!

So how do people actually break crypto?

- Very rarely: major algorithmic improvement
  - Biggest one recently: progress on small characteristic discrete logarithms/pairings
- More commonly: non-provably secure schemes shown to be insecure
  - Several of the TLS attacks
  - Many legacy scheme still in use could be broken (e.g. PKCS#1v1.5 signatures?)
- Most importantly: implementation attacks!

So how do people actually break crypto?

- Very rarely: major algorithmic improvement
  - Biggest one recently: progress on small characteristic discrete logarithms/pairings
- More commonly: non-provably secure schemes shown to be insecure
  - Several of the TLS attacks
  - Many legacy scheme still in use could be broken (e.g. PKCS#1v1.5 signatures?)
- Most importantly: implementation attacks!

Black-box vs real-world security

- Consider the security of e.g. RSA signatures
- Traditional, “black-box” view of security:
  - the attacker, Alice, interacts with the signer, Bob
  - Alice sends Bob messages to sign, only gets the results of Bob’s computation (no other info about the computation leaks)
  - based on that, Alice tries to forge new signatures/extract info about Bob’s signing key
- Real-world security:
  - Bob is actually a smart card, say
  - Alice can measure all sorts of emanation from the card as it operates, or mess with it in various ways
  - all that extra information can be useful to break things!

Black-box vs real-world security

- Consider the security of e.g. RSA signatures
- Traditional, “black-box” view of security:
  - the attacker, Alice, interacts with the signer, Bob
  - Alice sends Bob messages to sign, only gets the results of Bob’s computation (no other info about the computation leaks)
  - based on that, Alice tries to forge new signatures/extract info about Bob’s signing key
- Real-world security:
  - Bob is actually a smart card, say
  - Alice can measure all sorts of emanation from the card as it operates, or mess with it in various ways
  - all that extra information can be useful to break things!

Black-box vs real-world security

- Consider the security of e.g. RSA signatures
- Traditional, “black-box” view of security:
  - the attacker, Alice, interacts with the signer, Bob
  - Alice sends Bob messages to sign, only gets the results of Bob’s computation (no other info about the computation leaks)
  - based on that, Alice tries to forge new signatures/extract info about Bob’s signing key
- Real-world security:
  - Bob is actually a smart card, say
  - Alice can measure all sorts of emanation from the card as it operates, or mess with it in various ways
  - all that extra information can be useful to break things!

Implementation attacks

- To break a real-world crypto implementation, no need to play by the rules of black-box security
- In particular, provably secure schemes can be broken by bypassing the (usually black-box) security model
  - Remark: some attempts to also capture non black-box attacks in security proofs (e.g. leakage-resilient crypto...)
- These are implementation attacks

Implementation attacks

- To break a real-world crypto implementation, no need to play by the rules of black-box security
- In particular, provably secure schemes can be broken by bypassing the (usually black-box) security model
  - Remark: some attempts to also capture non black-box attacks in security proofs (e.g. leakage-resilient crypto...)
- These are implementation attacks

Implementation attacks

- To break a real-world crypto implementation, no need to play by the rules of black-box security
- In particular, provably secure schemes can be broken by bypassing the (usually black-box) security model
  - Remark: some attempts to also capture non black-box attacks in security proofs (e.g. leakage-resilient crypto...)
- These are implementation attacks

Various types of implementation attacks

- Correctness attacks: use the implementation as a black box, but send malformed/incorrect/invalid/malicious inputs
- Side-channel attacks: passive physical attacks, exploiting information leakage about the computation or the keys
- Fault attacks: active physical attacks, trying to extract secret information by tampering with the device to cause errors during the cryptographic computation

Various types of implementation attacks

- Correctness attacks: use the implementation as a black box, but send malformed/incorrect/invalid/malicious inputs
- Side-channel attacks: passive physical attacks, exploiting information leakage about the computation or the keys
- Fault attacks: active physical attacks, trying to extract secret information by tampering with the device to cause errors during the cryptographic computation

Various types of implementation attacks

- Correctness attacks: use the implementation as a black box, but send malformed/incorrect/invalid/malicious inputs
- Side-channel attacks: passive physical attacks, exploiting information leakage about the computation or the keys
- Fault attacks: active physical attacks, trying to extract secret information by tampering with the device to cause errors during the cryptographic computation

Outline

Introduction

- Implementation attacks
- Implementation attacks on lattice schemes

Physical attacks against BLISS

- The BLISS signature scheme
- Fault attack on the Gaussian sampling
- SCA on the rejection sampling

Conclusion and countermeasures

Towards postquantum cryptography

- Quantum computers would break all currently deployed public-key crypto: RSA, discrete logs, elliptic curves
- Agencies warn that we should prepare the transition to quantum-resistant crypto
  - NSA deprecating Suite B (elliptic curves)
  - NIST starting their postquantum competition
- In theory, plenty of known schemes are quantum-resistant
  - Some primitives achieved with codes, hash trees, multivariate crypto, knapsacks, isogenies…
  - Almost everything possible with lattices
- In practice, few actual implementations
  - Secure parameters often unclear
  - Concrete software/hardware implementation papers quite rare
  - Almost no consideration for implementation attacks
- Serious issue if we want practical postquantum crypto

Towards postquantum cryptography

- Quantum computers would break all currently deployed public-key crypto: RSA, discrete logs, elliptic curves
- Agencies warn that we should prepare the transition to quantum-resistant crypto
    - NSA deprecating Suite B (elliptic curves)
    - NIST starting their postquantum competition
- In theory, plenty of known schemes are quantum-resistant
    - Some primitives achieved with codes, hash trees, multivariate crypto, knapsacks, isogenies...
    - Almost everything possible with lattices
- In practice, few actual implementations
    - Secure parameters often unclear
    - Concrete software/hardware implementation papers quite rare
    - Almost no consideration for implementation attacks
- Serious issue if we want practical postquantum crypto

Towards postquantum cryptography

- Quantum computers would break all currently deployed public-key crypto: RSA, discrete logs, elliptic curves
- Agencies warn that we should prepare the transition to quantum-resistant crypto
    - NSA deprecating Suite B (elliptic curves)
    - NIST starting their postquantum competition
- In theory, plenty of known schemes are quantum-resistant
    - Some primitives achieved with codes, hash trees, multivariate crypto, knapsacks, isogenies...
    - Almost everything possible with lattices
- In practice, few actual implementations
    - Secure parameters often unclear
    - Concrete software/hardware implementation papers quite rare
    - Almost no consideration for implementation attacks
- Serious issue if we want practical postquantum crypto

Towards postquantum cryptography

- Quantum computers would break all currently deployed public-key crypto: RSA, discrete logs, elliptic curves
- Agencies warn that we should prepare the transition to quantum-resistant crypto
  - NSA deprecating Suite B (elliptic curves)
  - NIST starting their postquantum competition
- In theory, plenty of known schemes are quantum-resistant
  - Some primitives achieved with codes, hash trees, multivariate crypto, knapsacks, isogenies...
  - Almost everything possible with lattices
- In practice, few actual implementations
  - Secure parameters often unclear
  - Concrete software/hardware implementation papers quite rare
  - Almost no consideration for implementation attacks
- Serious issue if we want practical postquantum crypto

Towards postquantum cryptography

- Quantum computers would break all currently deployed public-key crypto: RSA, discrete logs, elliptic curves
- Agencies warn that we should prepare the transition to quantum-resistant crypto
  - NSA deprecating Suite B (elliptic curves)
  - NIST starting their postquantum competition
- In theory, plenty of known schemes are quantum-resistant
  - Some primitives achieved with codes, hash trees, multivariate crypto, knapsacks, isogenies...
  - Almost everything possible with lattices
- In practice, few actual implementations
  - Secure parameters often unclear
  - Concrete software/hardware implementation papers quite rare
  - Almost no consideration for implementation attacks
- Serious issue if we want practical postquantum crypto

11/34 ©2017 NTT Secure Platform Laboratories

Implementations of lattice-based schemes (I)

- Implementation work on lattice-based crypto is limited and mostly academic
- A number of papers describing implementations of inefficient schemes
    - Encryption: implementation of Lindner–Peikert (CHES'12, plaintexts of several MBs)
    - Signatures: implementation of GPV (SAC'13, keys of dozen MBs)
    - Other primitives: a few papers about ID-schemes, homomorphic encryption, etc.

Implementations of lattice-based schemes (I)

- Implementation work on lattice-based crypto is limited and mostly academic
- A number of papers describing implementations of inefficient schemes
    - Encryption: implementation of Lindner–Peikert (CHES'12, plaintexts of several MBs)
    - Signatures: implementation of GPV (SAC'13, keys of dozen MBs)
    - Other primitives: a few papers about ID-schemes, homomorphic encryption, etc.

Implementations of lattice-based schemes (II)

- One scheme has “industry” backing and quite a bit of code: NTRU
  - NTRUEncrypt is an ANSI standard, and believed to be okay
  - NTRUSign is a trainwreck that has been patched and broken many times
- In terms of practical schemes, other than NTRU, main efforts on signatures
  - GLP: improvement of Lyubashevsky signatures, efficient in SW and HW (CHES’12)
  - BLISS: improvement of GPL, even better (CRYPTO’13, CHES’14)
  - DLP: hash-and-sign scheme using GPV sampling on NTRU lattices (AC’14)
  - A few others: PASSSign (ACNS’14), TESLA (AFRICACRYPT’16), etc.

Implementations of lattice-based schemes (II)

- One scheme has “industry” backing and quite a bit of code: NTRU
  - NTRUEncrypt is an ANSI standard, and believed to be okay
  - NTRUSign is a trainwreck that has been patched and broken many times
- In terms of practical schemes, other than NTRU, main efforts on signatures
  - GLP: improvement of Lyubashevsky signatures, efficient in SW and HW (CHES’12)
  - BLISS: improvement of GPL, even better (CRYPTO’13, CHES’14)
  - DLP: hash-and-sign scheme using GPV sampling on NTRU lattices (AC’14)
  - A few others: PASSSign (ACNS’14), TESLA (AFRICACRYPT’16), etc.

Outline

Introduction

- Implementation attacks
- Implementation attacks on lattice schemes

Physical attacks against BLISS

- The BLISS signature scheme
- Fault attack on the Gaussian sampling
- SCA on the rejection sampling

Conclusion and countermeasures

BLISS: the basics

- One of the top contenders for postquantum signatures
- Introduced by Ducas, Durmus, Lepoint and Lyubashevsky at CRYPTO'13
- Implementations on various platforms: desktop computers, microcontrollers/smartcards, FPGAs
- Deployed in the VPN library strongSwan

BLISS: the basics

- One of the top contenders for postquantum signatures
- Introduced by Ducas, Durmus, Lepoint and Lyubashevsky at CRYPTO'13
- Implementations on various platforms: desktop computers, microcontrollers/smartcards, FPGAs
- Deployed in the VPN library strongSwan

BLISS: the basics

- One of the top contenders for postquantum signatures
- Introduced by Ducas, Durmus, Lepoint and Lyubashevsky at CRYPTO'13
- Implementations on various platforms: desktop computers, microcontrollers/smartcards, FPGAs
- Deployed in the VPN library strongSwan

BLISS: the basics

- One of the top contenders for postquantum signatures
- Introduced by Ducas, Durmus, Lepoint and Lyubashevsky at CRYPTO'13
- Implementations on various platforms: desktop computers, microcontrollers/smartcards, FPGAs
- Deployed in the VPN library strongSwan

BLISS: signing and verification keys

- Works in the cyclotomic ring $R = \mathbb{Z}[\mathbf{x}]/(x^n + 1)$, $n = 512$
- Computations modulo the prime $q = 12289$
- Secret key: random sparse $\mathbf{s}_1, \mathbf{s}_2 \in R$ with coefficients in $\{-1, 0, 1\}$
- Verification key: $\mathbf{a} = -\mathbf{s}_2/\mathbf{s}_1 \bmod q$
  - restart if $\mathbf{s}_1$ not invertible

BLISS: signing and verification keys

- Works in the cyclotomic ring $R = \mathbb{Z}[\mathbf{x}]/(x^n + 1)$, $n = 512$
- Computations modulo the prime $q = 12289$
- Secret key: random sparse $\mathbf{s}_1, \mathbf{s}_2 \in R$ with coefficients in $\{-1, 0, 1\}$
- Verification key: $\mathbf{a} = -\mathbf{s}_2/\mathbf{s}_1 \bmod q$
  - restart if $\mathbf{s}_1$ not invertible

BLISS: signing and verification keys

- Works in the cyclotomic ring $R = \mathbb{Z}[\mathbf{x}]/(x^n + 1)$, $n = 512$
- Computations modulo the prime $q = 12289$
- Secret key: random sparse $\mathbf{s}_1, \mathbf{s}_2 \in R$ with coefficients in $\{-1, 0, 1\}$
- Verification key: $\mathbf{a} = -\mathbf{s}_2/\mathbf{s}_1 \bmod q$
  - restart if $\mathbf{s}_1$ not invertible

BLISS: signing and verification keys

- Works in the cyclotomic ring $R = \mathbb{Z}[\mathbf{x}]/(x^n + 1)$, $n = 512$
- Computations modulo the prime $q = 12289$
- Secret key: random sparse $\mathbf{s}_1, \mathbf{s}_2 \in R$ with coefficients in $\{-1, 0, 1\}$
- Verification key: $\mathbf{a} = -\mathbf{s}_2/\mathbf{s}_1 \bmod q$
  - restart if $\mathbf{s}_1$ not invertible

BLISS: signature (simplified)

```
function $\textsc{Sign}(\mu, pk = \mathbf{a},\ sk = \mathbf{S} = (\mathbf{s}_1, \mathbf{s}_2))$
    $\mathbf{y}_1, \mathbf{y}_2 \leftarrow D^n_{\mathbb{Z},\sigma}$                ▷ Gaussian sampling
    $\mathbf{c} \leftarrow H(\mathbf{a} \cdot \mathbf{y}_1 + \mathbf{y}_2, \mu)$                ▷ special hashing
    choose a random bit $b$
    $\mathbf{z}_1 \leftarrow \mathbf{y}_1 + (-1)^b \mathbf{s}_1 \mathbf{c}$
    $\mathbf{z}_2 \leftarrow \mathbf{y}_2 + (-1)^b \mathbf{s}_2 \mathbf{c}$
    continue with probability
$1/\big(M \exp(-\|\mathbf{Sc}\|^2/(2\sigma^2)) \cosh(\langle \mathbf{z}, \mathbf{Sc} \rangle/\sigma^2\big)$ otherwise restart
    $\mathbf{z}_2^\dagger \leftarrow \textsc{Compress}(\mathbf{z}_2)$
    return $(\mathbf{z}_1, \mathbf{z}_2^\dagger, \mathbf{c})$
end function
```

BLISS: signature (simplified)

```
function SIGN($\mu, pk = \mathbf{a},\ sk = \mathbf{S} = (\mathbf{s}_1, \mathbf{s}_2)$)
    $\mathbf{y}_1, \mathbf{y}_2 \leftarrow D^n_{\mathbb{Z},\sigma}$                      ▷ Gaussian sampling
    $\mathbf{c} \leftarrow H(\mathbf{a} \cdot \mathbf{y}_1 + \mathbf{y}_2, \mu)$                     ▷ special hashing
    choose a random bit $b$
    $\mathbf{z}_1 \leftarrow \mathbf{y}_1 + (-1)^b \mathbf{s}_1 \mathbf{c}$
    $\mathbf{z}_2 \leftarrow \mathbf{y}_2 + (-1)^b \mathbf{s}_2 \mathbf{c}$
    continue with probability
$1/\big(M \exp(-\|\mathbf{Sc}\|^2/(2\sigma^2)) \cosh(\langle \mathbf{z}, \mathbf{Sc} \rangle/\sigma^2\big)$ otherwise restart
    $\mathbf{z}_2^\dagger \leftarrow$ COMPRESS($\mathbf{z}_2$)
    return $(\mathbf{z}_1, \mathbf{z}_2^\dagger, \mathbf{c})$
end function
```

BLISS: signature (simplified)

```
function SIGN($\mu, pk = \mathbf{a},\ sk = \mathbf{S} = (\mathbf{s}_1, \mathbf{s}_2)$)
    $\mathbf{y}_1, \mathbf{y}_2 \leftarrow D^n_{\mathbb{Z},\sigma}$                      ▷ Gaussian sampling
    $\mathbf{c} \leftarrow H(\mathbf{a} \cdot \mathbf{y}_1 + \mathbf{y}_2, \mu)$                     ▷ special hashing
    choose a random bit $b$
    $\mathbf{z}_1 \leftarrow \mathbf{y}_1 + (-1)^b \mathbf{s}_1 \mathbf{c}$
    $\mathbf{z}_2 \leftarrow \mathbf{y}_2 + (-1)^b \mathbf{s}_2 \mathbf{c}$
    continue with probability
$1/\big(M \exp(-\|\mathbf{Sc}\|^2/(2\sigma^2)) \cosh(\langle \mathbf{z}, \mathbf{Sc} \rangle/\sigma^2\big)$ otherwise restart
    $\mathbf{z}_2^\dagger \leftarrow$ COMPRESS($\mathbf{z}_2$)
    return $(\mathbf{z}_1, \mathbf{z}_2^\dagger, \mathbf{c})$
end function
```

BLISS: signature (simplified)

```
function $\textsc{Sign}(\mu, pk = \mathbf{a},\ sk = \mathbf{S} = (\mathbf{s}_1, \mathbf{s}_2))$
    $\mathbf{y}_1, \mathbf{y}_2 \leftarrow D^n_{\mathbb{Z},\sigma}$                          ▷ Gaussian sampling
    $\mathbf{c} \leftarrow H(\mathbf{a} \cdot \mathbf{y}_1 + \mathbf{y}_2, \mu)$                     ▷ special hashing
    choose a random bit $b$
    $\mathbf{z}_1 \leftarrow \mathbf{y}_1 + (-1)^b \mathbf{s}_1 \mathbf{c}$
    $\mathbf{z}_2 \leftarrow \mathbf{y}_2 + (-1)^b \mathbf{s}_2 \mathbf{c}$
    continue with probability
$1/\big(M \exp(-\|\mathbf{Sc}\|^2/(2\sigma^2))\cosh(\langle \mathbf{z}, \mathbf{Sc}\rangle/\sigma^2\big)$ otherwise restart
    $\mathbf{z}_2^\dagger \leftarrow \textsc{Compress}(\mathbf{z}_2)$
    return $(\mathbf{z}_1, \mathbf{z}_2^\dagger, \mathbf{c})$
end function
```

BLISS: signature (simplified)

function $\textsc{Sign}(\mu, pk = \mathbf{a},\ sk = \mathbf{S} = (\mathbf{s}_1, \mathbf{s}_2))$
    $\mathbf{y}_1, \mathbf{y}_2 \leftarrow D^n_{\mathbb{Z},\sigma}$ ▷ Gaussian sampling
    $\mathbf{c} \leftarrow H(\mathbf{a} \cdot \mathbf{y}_1 + \mathbf{y}_2, \mu)$ ▷ special hashing
    choose a random bit b
    $\mathbf{z}_1 \leftarrow \mathbf{y}_1 + (-1)^b \mathbf{s}_1 \mathbf{c}$
    $\mathbf{z}_2 \leftarrow \mathbf{y}_2 + (-1)^b \mathbf{s}_2 \mathbf{c}$
    continue with probability
$1/\big(M \exp(-\|\mathbf{Sc}\|^2/(2\sigma^2))\cosh(\langle \mathbf{z}, \mathbf{Sc}\rangle/\sigma^2\big)$ otherwise restart
    $\mathbf{z}_2^\dagger \leftarrow \textsc{Compress}(\mathbf{z}_2)$
    return $(\mathbf{z}_1, \mathbf{z}_2^\dagger, \mathbf{c})$
end function

BLISS: verification

To check if $(\mathbf{z}_1, \mathbf{z}_2^\dagger, \mathbf{c})$ is a valid signature:

1. Uncompress $\mathbf{z}_2^\dagger$ to essentially get $\mathbf{z}_2$
2. Check if $\mathbf{z}_1, \mathbf{z}_2, \mathbf{c}$ are small enough
3. Compute $\mathbf{u} = \mathbf{a} \cdot \mathbf{z}_1 + \mathbf{z}_2$; it satisfies:

$$\begin{aligned} \mathbf{u} &= \mathbf{a} \cdot (\mathbf{y}_1 + (-1)^b \mathbf{s}_1 \mathbf{c}) + (\mathbf{y}_2 + (-1)^b \mathbf{s}_2 \mathbf{c}) \\ &= (\mathbf{a}\mathbf{y}_1 + \mathbf{y}_2) + (-1)^b (\mathbf{a}\mathbf{s}_1 + \mathbf{s}_2) \qquad = \mathbf{a}\mathbf{y}_1 + \mathbf{y}_2 \pmod q \end{aligned}$$

   since $\mathbf{a} = -\mathbf{s}_2/\mathbf{s}_1 \bmod q$

4. Check whether $H(\mathbf{u}) \overset{?}{=} \mathbf{c}$

Works even with approximate decompression, because H depends only on the most significant bits of its input

BLISS: parameters

- Parameters proposed by Ducas et al. for 128-bit security (BLISS–I)
  - $n = 512,\ q = 12289$
  - $\delta = 0.3$ (density of $\mathbf{s}_1, \mathbf{s}_2$)
  - $\sigma = 215$ (std. dev. of $\mathbf{y}_1, \mathbf{y}_2$)
  - $\kappa = 23$ (number of 1's in $\mathbf{c}$)

Outline

Introduction

- Implementation attacks
- Implementation attacks on lattice schemes

Physical attacks against BLISS

- The BLISS signature scheme
- Fault attack on the Gaussian sampling
- SCA on the rejection sampling

Conclusion and countermeasures

Attacking $\mathbf{y}$

- The ring element $\mathbf{y}_1$, which acts as additive mask in the relation:

$$\mathbf{z}_1 \equiv \mathbf{y}_1 + (-1)^b \mathbf{s}_1 \mathbf{c} \pmod q$$

  is sampled according to a discrete Gaussian
- Sampling carried out coefficient by coefficient
- Idea of the attack: use fault injection to abort the sampling early, so that a faulty signature will be generated with a low-degree $\mathbf{y}_1$
- Can be done by attacking the branching test of the loop (voltage spike, clock variation...), or the contents of the loop counter (lasers, x-rays...)

Attacking y

- The ring element $\mathbf{y}_1$, which acts as additive mask in the relation:
$$\mathbf{z}_1 \equiv \mathbf{y}_1 + (-1)^b \mathbf{s}_1 \mathbf{c} \pmod q$$
is sampled according to a discrete Gaussian
- Sampling carried out coefficient by coefficient
- Idea of the attack: use fault injection to abort the sampling early, so that a faulty signature will be generated with a low-degree $\mathbf{y}_1$
- Can be done by attacking the branching test of the loop (voltage spike, clock variation...), or the contents of the loop counter (lasers, x-rays...)

Attacking y

- The ring element $\mathbf{y}_1$, which acts as additive mask in the relation:
$$\mathbf{z}_1 \equiv \mathbf{y}_1 + (-1)^b \mathbf{s}_1 \mathbf{c} \pmod q$$
is sampled according to a discrete Gaussian
- Sampling carried out coefficient by coefficient
- Idea of the attack: use fault injection to abort the sampling early, so that a faulty signature will be generated with a low-degree $\mathbf{y}_1$
- Can be done by attacking the branching test of the loop (voltage spike, clock variation...), or the contents of the loop counter (lasers, x-rays...)

Attacking y

- The ring element $\mathbf{y}_1$, which acts as additive mask in the relation:

$$\mathbf{z}_1 \equiv \mathbf{y}_1 + (-1)^b \mathbf{s}_1 \mathbf{c} \pmod q$$

  is sampled according to a discrete Gaussian
- Sampling carried out coefficient by coefficient
- Idea of the attack: use fault injection to abort the sampling early, so that a faulty signature will be generated with a low-degree $\mathbf{y}_1$
- Can be done by attacking the branching test of the loop (voltage spike, clock variation...), or the contents of the loop counter (lasers, x-rays...)

Attack details (I)

- So let's say we get a signature generated with $\mathbf{y}_1$ of degree $m \ll n$
- If $\mathbf{c}$ is invertible (probability around $(1-1/q)^n \approx 96\%$), we can compute:

$$\mathbf{v} = \mathbf{c}^{-1}\mathbf{z}_1 \equiv \mathbf{c}^{-1}\mathbf{y}_1 + (-1)^b \mathbf{s}_1 \pmod q$$

- WLOG, $b = 0$ (equivalent keys)
- Since $\mathbf{s}_1$ is very short, $\mathbf{v}$ very close to the lattice L generated by $q\mathbb{Z}^n$ and $\mathbf{w}_i = \mathbf{c}^{-1}\mathbf{x}^i$, $i = 0, \ldots, m$
- L of dimension n: too large to apply lattice reduction
- However, we have the same relation on arbitrary subset of coefficients: we can reduce the dimension

Attack details (I)

- So let's say we get a signature generated with $\mathbf{y}_1$ of degree $m \ll n$
- If $\mathbf{c}$ is invertible (probability around $(1 - 1/q)^n \approx 96\%$), we can compute:

$$\mathbf{v} = \mathbf{c}^{-1}\mathbf{z}_1 \equiv \mathbf{c}^{-1}\mathbf{y}_1 + (-1)^b\mathbf{s}_1 \pmod q$$

- WLOG, $b = 0$ (equivalent keys)
- Since $\mathbf{s}_1$ is very short, $\mathbf{v}$ very close to the lattice L generated by $q\mathbb{Z}^n$ and $\mathbf{w}_i = \mathbf{c}^{-1}\mathbf{x}^i$, $i = 0, \ldots, m$
- L of dimension n: too large to apply lattice reduction
- However, we have the same relation on arbitrary subset of coefficients: we can reduce the dimension

Attack details (I)

- So let's say we get a signature generated with $\mathbf{y}_1$ of degree $m \ll n$
- If $\mathbf{c}$ is invertible (probability around $(1 - 1/q)^n \approx 96\%$), we can compute:

$$\mathbf{v} = \mathbf{c}^{-1}\mathbf{z}_1 \equiv \mathbf{c}^{-1}\mathbf{y}_1 + (-1)^b\mathbf{s}_1 \pmod q$$

- WLOG, $b = 0$ (equivalent keys)
- Since $\mathbf{s}_1$ is very short, $\mathbf{v}$ very close to the lattice L generated by $q\mathbb{Z}^n$ and $\mathbf{w}_i = \mathbf{c}^{-1}\mathbf{x}^i$, $i = 0, \ldots, m$
- L of dimension n: too large to apply lattice reduction
- However, we have the same relation on arbitrary subset of coefficients: we can reduce the dimension

Attack details (I)

- So let's say we get a signature generated with $\mathbf{y}_1$ of degree $m \ll n$
- If $\mathbf{c}$ is invertible (probability around $(1-1/q)^n \approx 96\%$), we can compute:

$$\mathbf{v} = \mathbf{c}^{-1}\mathbf{z}_1 \equiv \mathbf{c}^{-1}\mathbf{y}_1 + (-1)^b\mathbf{s}_1 \pmod{q}$$

- WLOG, $b = 0$ (equivalent keys)
- Since $\mathbf{s}_1$ is very short, $\mathbf{v}$ very close to the lattice L generated by $q\mathbb{Z}^n$ and $\mathbf{w}_i = \mathbf{c}^{-1}\mathbf{x}^i$, $i = 0, \ldots, m$
- L of dimension n: too large to apply lattice reduction
- However, we have the same relation on arbitrary subset of coefficients: we can reduce the dimension

Attack details (I)

- So let's say we get a signature generated with $\mathbf{y}_1$ of degree $m \ll n$
- If $\mathbf{c}$ is invertible (probability around $(1-1/q)^n \approx 96\%$), we can compute:

$$\mathbf{v} = \mathbf{c}^{-1}\mathbf{z}_1 \equiv \mathbf{c}^{-1}\mathbf{y}_1 + (-1)^b\mathbf{s}_1 \pmod{q}$$

- WLOG, $b = 0$ (equivalent keys)
- Since $\mathbf{s}_1$ is very short, $\mathbf{v}$ very close to the lattice L generated by $q\mathbb{Z}^n$ and $\mathbf{w}_i = \mathbf{c}^{-1}\mathbf{x}^i$, $i = 0, \ldots, m$
- L of dimension n: too large to apply lattice reduction
- However, we have the same relation on arbitrary subset of coefficients: we can reduce the dimension

Attack details (I)

- So let's say we get a signature generated with $\mathbf{y}_1$ of degree $m \ll n$
- If $\mathbf{c}$ is invertible (probability around $(1-1/q)^n \approx 96\%$), we can compute:

$$\mathbf{v} = \mathbf{c}^{-1}\mathbf{z}_1 \equiv \mathbf{c}^{-1}\mathbf{y}_1 + (-1)^b \mathbf{s}_1 \pmod q$$

- WLOG, $b = 0$ (equivalent keys)
- Since $\mathbf{s}_1$ is very short, $\mathbf{v}$ very close to the lattice L generated by $q\mathbb{Z}^n$ and $\mathbf{w}_i = \mathbf{c}^{-1}\mathbf{x}^i$, $i = 0, \dots, m$
- L of dimension n: too large to apply lattice reduction
- However, we have the same relation on arbitrary subset of coefficients: we can reduce the dimension

Attack details (II)

- More precisely, fix a subset $I \subset \{0, \dots, n-1\}$ of ℓ indices, and let $\varphi_I : \mathbb{Z}^n \to \mathbb{Z}^I$ be the obvious projection
- $\varphi_I(\mathbf{v})$ is close to the lattice generated by $\varphi_I(\mathbf{w}_i)$ and $q\mathbb{Z}^I$, and if ℓ is large enough, the difference should be $\varphi_I(\mathbf{s}_1)$.
- Solve this close vector problem using Babai nearest plane algorithm. Condition on ℓ to recover $\varphi_I(\mathbf{s}_1)$:

$$\ell + 1 \gtrsim \frac{m + 2 + \frac{\log\sqrt{\delta_1 + 4\delta_2}}{\log q}}{1 - \frac{\log\sqrt{2\pi e(\delta_1 + 4\delta_2)}}{\log q}}$$

- For BLISS–I and BLISS–II, this says $\ell \approx 1.09 \cdot m$
- In practice: works fine with LLL for $m \lesssim 60$ and with BKZ with $m \lesssim 100$
- Just apply the attack for several choices of I to recover all of $\mathbf{s}_1$, and subsequently $\mathbf{s}_2$: full key recovery with one fauly sig.!

Attack details (II)

- More precisely, fix a subset $I \subset \{0, \ldots, n-1\}$ of ℓ indices, and let $\varphi_I \colon \mathbb{Z}^n \to \mathbb{Z}^I$ be the obvious projection
- $\varphi_I(\mathbf{v})$ is close to the lattice generated by $\varphi_I(\mathbf{w}_i)$ and $q\mathbb{Z}^I$, and if ℓ is large enough, the difference should be $\varphi_I(\mathbf{s}_1)$.
- Solve this close vector problem using Babai nearest plane algorithm. Condition on ℓ to recover $\varphi_I(\mathbf{s}_1)$:

$$\ell + 1 \gtrsim \frac{m + 2 + \frac{\log\sqrt{\delta_1 + 4\delta_2}}{\log q}}{1 - \frac{\log\sqrt{2\pi e(\delta_1 + 4\delta_2)}}{\log q}}$$

- For BLISS–I and BLISS–II, this says $\ell \approx 1.09 \cdot m$
- In practice: works fine with LLL for $m \lesssim 60$ and with BKZ with $m \lesssim 100$
- Just apply the attack for several choices of I to recover all of $\mathbf{s}_1$, and subsequently $\mathbf{s}_2$: full key recovery with one fauly sig.!

Attack details (II)

- More precisely, fix a subset $I \subset \{0, \ldots, n-1\}$ of ℓ indices, and let $\varphi_I \colon \mathbb{Z}^n \to \mathbb{Z}^I$ be the obvious projection
- $\varphi_I(\mathbf{v})$ is close to the lattice generated by $\varphi_I(\mathbf{w}_i)$ and $q\mathbb{Z}^I$, and if ℓ is large enough, the difference should be $\varphi_I(\mathbf{s}_1)$.
- Solve this close vector problem using Babai nearest plane algorithm. Condition on ℓ to recover $\varphi_I(\mathbf{s}_1)$:

$$\ell + 1 \gtrsim \frac{m + 2 + \frac{\log\sqrt{\delta_1 + 4\delta_2}}{\log q}}{1 - \frac{\log\sqrt{2\pi e(\delta_1 + 4\delta_2)}}{\log q}}$$

- For BLISS–I and BLISS–II, this says $\ell \approx 1.09 \cdot m$
- In practice: works fine with LLL for $m \lesssim 60$ and with BKZ with $m \lesssim 100$
- Just apply the attack for several choices of I to recover all of $\mathbf{s}_1$, and subsequently $\mathbf{s}_2$: full key recovery with one fauly sig.!

Attack details (II)

- More precisely, fix a subset $I \subset \{0, \ldots, n-1\}$ of ℓ indices, and let $\varphi_I : \mathbb{Z}^n \to \mathbb{Z}^I$ be the obvious projection
- $\varphi_I(\mathbf{v})$ is close to the lattice generated by $\varphi_I(\mathbf{w}_i)$ and $q\mathbb{Z}^I$, and if ℓ is large enough, the difference should be $\varphi_I(\mathbf{s}_1)$.
- Solve this close vector problem using Babai nearest plane algorithm. Condition on ℓ to recover $\varphi_I(\mathbf{s}_1)$:

$$\ell + 1 \gtrsim \frac{m + 2 + \frac{\log\sqrt{\delta_1 + 4\delta_2}}{\log q}}{1 - \frac{\log\sqrt{2\pi e(\delta_1 + 4\delta_2)}}{\log q}}$$

- For BLISS–I and BLISS–II, this says $\ell \approx 1.09 \cdot m$
- In practice: works fine with LLL for $m \lesssim 60$ and with BKZ with $m \lesssim 100$
- Just apply the attack for several choices of I to recover all of $\mathbf{s}_1$, and subsequently $\mathbf{s}_2$: full key recovery with one fauly sig.!

Attack details (II)

- More precisely, fix a subset $I \subset \{0, \ldots, n-1\}$ of ℓ indices, and let $\varphi_I : \mathbb{Z}^n \to \mathbb{Z}^I$ be the obvious projection
- $\varphi_I(\mathbf{v})$ is close to the lattice generated by $\varphi_I(\mathbf{w}_i)$ and $q\mathbb{Z}^I$, and if ℓ is large enough, the difference should be $\varphi_I(\mathbf{s}_1)$.
- Solve this close vector problem using Babai nearest plane algorithm. Condition on ℓ to recover $\varphi_I(\mathbf{s}_1)$:

$$\ell + 1 \gtrsim \frac{m + 2 + \frac{\log\sqrt{\delta_1 + 4\delta_2}}{\log q}}{1 - \frac{\log\sqrt{2\pi e(\delta_1 + 4\delta_2)}}{\log q}}$$

- For BLISS–I and BLISS–II, this says $\ell \approx 1.09 \cdot m$
- In practice: works fine with LLL for $m \lesssim 60$ and with BKZ with $m \lesssim 100$
- Just apply the attack for several choices of I to recover all of $\mathbf{s}_1$, and subsequently $\mathbf{s}_2$: full key recovery with one fauly sig.!

Attack details (II)

- More precisely, fix a subset $I \subset \{0, \ldots, n-1\}$ of ℓ indices, and let $\varphi_I : \mathbb{Z}^n \to \mathbb{Z}^I$ be the obvious projection
- $\varphi_I(\mathbf{v})$ is close to the lattice generated by $\varphi_I(\mathbf{w}_i)$ and $q\mathbb{Z}^I$, and if ℓ is large enough, the difference should be $\varphi_I(\mathbf{s}_1)$.
- Solve this close vector problem using Babai nearest plane algorithm. Condition on ℓ to recover $\varphi_I(\mathbf{s}_1)$:

$$\ell + 1 \gtrsim \frac{m + 2 + \frac{\log \sqrt{\delta_1 + 4\delta_2}}{\log q}}{1 - \frac{\log \sqrt{2\pi e(\delta_1 + 4\delta_2)}}{\log q}}$$

- For BLISS–I and BLISS–II, this says $\ell \approx 1.09 \cdot m$
- In practice: works fine with LLL for $m \lesssim 60$ and with BKZ with $m \lesssim 100$
- Just apply the attack for several choices of I to recover all of $\mathbf{s}_1$, and subsequently $\mathbf{s}_2$: full key recovery with one fauly sig.!

Implementation results

Fault after iteration number $m =$	5	10	20	40	80	100
Theoretical minimum dimension $\ell_{\min}$	6	11	22	44	88	110
Dimension ℓ in our experiment	6	12	24	50	110	150
Lattice reduction algorithm	LLL	LLL	LLL	BKZ–20	BKZ–25	BKZ–25
Success probability (%)	99	100	100	100	100	98
Avg. CPU time to recover ℓ coeffs. (s)	0.005	0.022	0.23	7.3	941	33655
Avg. CPU time for full key recovery	0.5 s	1 s	5 s	80 s	80 min	38 h

Outline

Introduction

Implementation attacks

Implementation attacks on lattice schemes

Physical attacks against BLISS

The BLISS signature scheme

Fault attack on the Gaussian sampling

SCA on the rejection sampling

Conclusion and countermeasures

Attack overview

- Attack on the rejection sampling
  - cornerstone of BLISS security/efficiency
- Straightforward implementation of rejection sampling would be inefficient: use optimized rejection algorithm
- Idea of the optimization: iterated Bernoulli trials on the bits of $\|\mathbf{Sc}\|^2$
- Side-channel leakage: can read off $\|\mathbf{Sc}\|^2$ on SPA/SEMA trace!
- From a few of these: recover $\mathbf{s}_1 \cdot \bar{\mathbf{s}_1}$ ("relative norm" of the secret key)
- Then, algebraic number theory to retrieve $\mathbf{s}_1$

Attack overview

- Attack on the rejection sampling
  - cornerstone of BLISS security/efficiency
- Straightforward implementation of rejection sampling would be inefficient: use optimized rejection algorithm
- Idea of the optimization: iterated Bernoulli trials on the bits of $\|\mathbf{Sc}\|^2$
- Side-channel leakage: can read off $\|\mathbf{Sc}\|^2$ on SPA/SEMA trace!
- From a few of these: recover $\mathbf{s_1} \cdot \bar{\mathbf{s_1}}$ ("relative norm" of the secret key)
- Then, algebraic number theory to retrieve $\mathbf{s}_1$

Attack overview

- Attack on the rejection sampling
  - cornerstone of BLISS security/efficiency
- Straightforward implementation of rejection sampling would be inefficient: use optimized rejection algorithm
- Idea of the optimization: iterated Bernoulli trials on the bits of $\|\mathbf{Sc}\|^2$
- Side-channel leakage: can read off $\|\mathbf{Sc}\|^2$ on SPA/SEMA trace!
- From a few of these: recover $\mathbf{s_1} \cdot \bar{\mathbf{s_1}}$ ("relative norm" of the secret key)
- Then, algebraic number theory to retrieve $\mathbf{s}_1$

Attack overview

- Attack on the rejection sampling
  - cornerstone of BLISS security/efficiency
- Straightforward implementation of rejection sampling would be inefficient: use optimized rejection algorithm
- Idea of the optimization: iterated Bernoulli trials on the bits of $\|\mathbf{Sc}\|^2$
- Side-channel leakage: can read off $\|\mathbf{Sc}\|^2$ on SPA/SEMA trace!
- From a few of these: recover $\mathbf{s_1} \cdot \bar{\mathbf{s}}_1$ (“relative norm” of the secret key)
- Then, algebraic number theory to retrieve $\mathbf{s}_1$

Attack overview

- Attack on the rejection sampling
  - cornerstone of BLISS security/efficiency
- Straightforward implementation of rejection sampling would be inefficient: use optimized rejection algorithm
- Idea of the optimization: iterated Bernoulli trials on the bits of $\|\mathbf{Sc}\|^2$
- Side-channel leakage: can read off $\|\mathbf{Sc}\|^2$ on SPA/SEMA trace!
- From a few of these: recover $\mathbf{s_1} \cdot \bar{\mathbf{s}}_1$ (“relative norm” of the secret key)
- Then, algebraic number theory to retrieve $\mathbf{s}_1$

Attack overview

- Attack on the rejection sampling
  - cornerstone of BLISS security/efficiency
- Straightforward implementation of rejection sampling would be inefficient: use optimized rejection algorithm
- Idea of the optimization: iterated Bernoulli trials on the bits of $\|\mathbf{Sc}\|^2$
- Side-channel leakage: can read off $\|\mathbf{Sc}\|^2$ on SPA/SEMA trace!
- From a few of these: recover $\mathbf{s_1} \cdot \bar{\mathbf{s_1}}$ ("relative norm" of the secret key)
- Then, algebraic number theory to retrieve $\mathbf{s}_1$

BLISS rejection sampling

```
function SampleBernExp($x$)
    for $i = 0$ to $\ell - 1$ do
        if $x_i = 1$ then
            Sample $a \leftarrow \mathscr{B}_{c_i}$
            if $a = 0$ then return 0
        end if
    end for
    return 1
end function    $\triangleright\ x = K - \|\mathbf{Sc}\|^2$
```

```
function SampleBernCosh($x$)
    Sample $a \leftarrow \mathscr{B}_{\exp(-x/f)}$
    if $a = 1$ then return 1
    Sample $b \leftarrow \mathscr{B}_{1/2}$
    if $b = 1$ then restart
    Sample $c \leftarrow \mathscr{B}_{\exp(-x/f)}$
    if $c = 1$ then restart
    return 0
end function    $\triangleright\ x = 2 \cdot \langle \mathbf{z}, \mathbf{Sc} \rangle$
```

Sampling algorithms for the distributions $\mathscr{B}_{\exp(-x/f)}$ and $\mathscr{B}_{1/\cosh(x/f)}$ ($c_i = 2^i/f$ precomputed)

BLISS rejection sampling

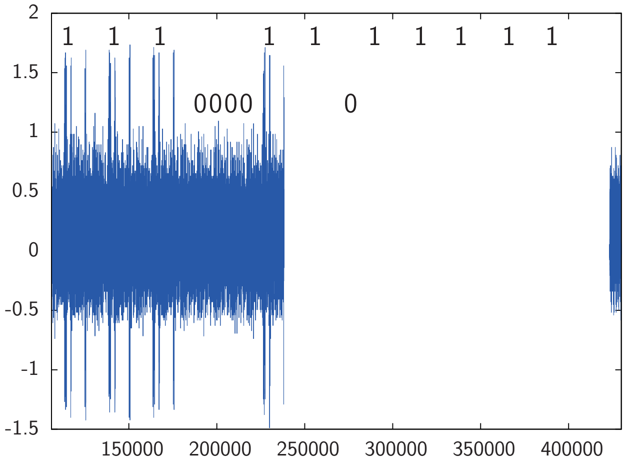
```
function SampleBernExp($x$)
    for $i = 0$ to $\ell - 1$ do
        if $x_i = 1$ then
            Sample $a \leftarrow \mathscr{B}_{c_i}$
            if $a = 0$ then return 0
        end if
    end for
    return 1
end function    ▷ $x = K - \|\mathbf{Sc}\|^2$
```

```
function SampleBernCosh($x$)
    Sample $a \leftarrow \mathscr{B}_{\exp(-x/f)}$
    if $a = 1$ then return 1
    Sample $b \leftarrow \mathscr{B}_{1/2}$
    if $b = 1$ then restart
    Sample $c \leftarrow \mathscr{B}_{\exp(-x/f)}$
    if $c = 1$ then restart
    return 0
end function    ▷ $x = 2 \cdot \langle \mathbf{z}, \mathbf{Sc} \rangle$
```

Sampling algorithms for the distributions $\mathscr{B}_{\exp(-x/f)}$ and $\mathscr{B}_{1/\cosh(x/f)}$ ($c_i = 2^i/f$ precomputed)

Experimental leakage

EMA trace of BLISS rejection sampling on 8-bit AVR for norm $\|\mathbf{Sc}\|^2 = 14404$. One reads the value: $K - \|\mathbf{Sc}\|^2 = 46539 - 14404 = 32135 = \overline{111110110000111}_2$



Exploiting the leakage

- From each trace, get:

$$\|\mathbf{Sc}\|^2 = \|\mathbf{s}_1 \cdot \mathbf{c}\|^2 + \|\mathbf{s}_2 \cdot \mathbf{c}\|^2$$

  for known **c**, different each time
- Linear equation on $\mathbf{s}_1 \cdot \bar{\mathbf{s}}_1$ and $\mathbf{s}_2 \cdot \bar{\mathbf{s}}_2$
- Collect $\approx 2 \times 256 = 512$ to recover the relative norms $\mathbf{s}_i \cdot \bar{\mathbf{s}}_i$
  - linear equations are independent w.h.p.
  - very efficient in practice
  - collecting 512 EM traces an easy task
- Going from $\mathbf{s}_1 \cdot \bar{\mathbf{s}}_1$ to $\mathbf{s}_1$: algebraic number theory (Howgrave-Graham–Szydlo)

Exploiting the leakage

- From each trace, get:

$$\|\mathbf{Sc}\|^2 = \|\mathbf{s}_1 \cdot \mathbf{c}\|^2 + \|\mathbf{s}_2 \cdot \mathbf{c}\|^2$$

  for known **c**, different each time
- Linear equation on $\mathbf{s}_1 \cdot \bar{\mathbf{s}}_1$ and $\mathbf{s}_2 \cdot \bar{\mathbf{s}}_2$
- Collect $\approx 2 \times 256 = 512$ to recover the relative norms $\mathbf{s}_i \cdot \bar{\mathbf{s}}_i$
  - linear equations are independent w.h.p.
  - very efficient in practice
  - collecting 512 EM traces an easy task
- Going from $\mathbf{s}_1 \cdot \bar{\mathbf{s}}_1$ to $\mathbf{s}_1$: algebraic number theory (Howgrave-Graham–Szydlo)

Exploiting the leakage

- From each trace, get:

$$\left\|\mathbf{Sc}\right\|^2 = \left\|\mathbf{s}_1 \cdot \mathbf{c}\right\|^2 + \left\|\mathbf{s}_2 \cdot \mathbf{c}\right\|^2$$

  for known $\mathbf{c}$, different each time
- Linear equation on $\mathbf{s}_1 \cdot \bar{\mathbf{s}}_1$ and $\mathbf{s}_2 \cdot \bar{\mathbf{s}}_2$
- Collect $\approx 2 \times 256 = 512$ to recover the relative norms $\mathbf{s}_i \cdot \bar{\mathbf{s}}_i$
  - linear equations are independent w.h.p.
  - very efficient in practice
  - collecting 512 EM traces an easy task
- Going from $\mathbf{s}_1 \cdot \bar{\mathbf{s}}_1$ to $\mathbf{s}_1$: algebraic number theory (Howgrave-Graham–Szydlo)

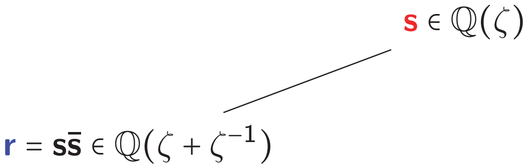
Exploiting the leakage

- From each trace, get:

$$\left\|\mathbf{Sc}\right\|^2 = \left\|\mathbf{s}_1 \cdot \mathbf{c}\right\|^2 + \left\|\mathbf{s}_2 \cdot \mathbf{c}\right\|^2$$

  for known $\mathbf{c}$, different each time
- Linear equation on $\mathbf{s}_1 \cdot \bar{\mathbf{s}}_1$ and $\mathbf{s}_2 \cdot \bar{\mathbf{s}}_2$
- Collect $\approx 2 \times 256 = 512$ to recover the relative norms $\mathbf{s}_i \cdot \bar{\mathbf{s}}_i$
  - linear equations are independent w.h.p.
  - very efficient in practice
  - collecting 512 EM traces an easy task
- Going from $\mathbf{s}_1 \cdot \bar{\mathbf{s}}_1$ to $\mathbf{s}_1$: algebraic number theory (Howgrave-Graham–Szydlo)

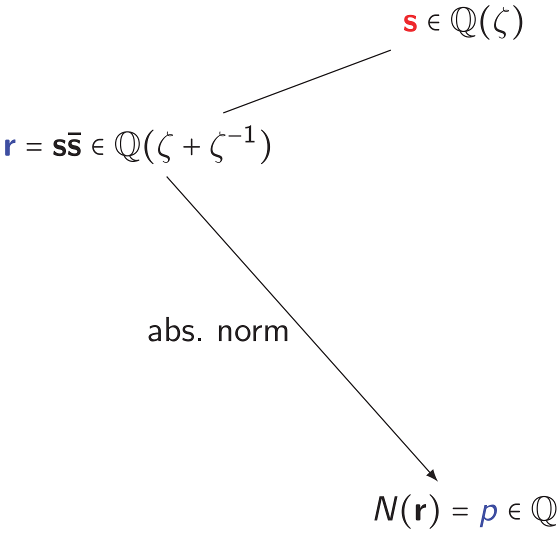
Howgrave-Graham–Szydlo in a nutshell



$\mathbf{s} \in \mathbb{Q}(\zeta)$

$\mathbf{r} = \mathbf{s}\bar{\mathbf{s}} \in \mathbb{Q}(\zeta + \zeta^{-1})$

Howgrave-Graham–Szydlo in a nutshell



$\mathbf{s} \in \mathbb{Q}(\zeta)$

$\mathbf{r} = \mathbf{s}\bar{\mathbf{s}} \in \mathbb{Q}(\zeta + \zeta^{-1})$

abs. norm

$N(\mathbf{r}) = p \in \mathbb{Q}$

Howgrave-Graham–Szydlo in a nutshell

$$\begin{tikzcd}
& \mathbf{s} \in \mathbb{Q}(\zeta) \arrow[dl, dash] & \\
\mathbf{r} = \mathbf{s}\bar{\mathbf{s}} \in \mathbb{Q}(\zeta + \zeta^{-1}) \arrow[ddr, "\text{abs. norm}"'] & & \\
& & p = \pi \cdot \bar{\pi} \in \mathbb{Q}(i) \\
& N(\mathbf{r}) = p \in \mathbb{Q} \arrow[ur, "\text{split}"'] &
\end{tikzcd}$$

Howgrave-Graham–Szydlo in a nutshell

$$\begin{tikzcd}
& \mathbf{s} \in \mathbb{Q}(\zeta) \supset \mathbf{s}R = \mathbf{r}R \cap \pi R \text{ or } \mathbf{r}R \cap \bar{p}iR \arrow[dl, dash] & \\
\mathbf{r} = \mathbf{s}\bar{\mathbf{s}} \in \mathbb{Q}(\zeta + \zeta^{-1}) \arrow[ddr, "\text{abs. norm}"'] & & \\
& & p = \pi \cdot \bar{\pi} \in \mathbb{Q}(i) \arrow[uul, "\text{lift}"'] \\
& N(\mathbf{r}) = p \in \mathbb{Q} \arrow[ur, "\text{split}"'] &
\end{tikzcd}$$

Howgrave-Graham–Szydlo in a nutshell

$$\begin{tikzcd}
 & \mathbf{s} \in \mathbb{Q}(\zeta) \supset \mathbf{s}R = \mathbf{r}R \cap \pi R \text{ or } \mathbf{r}R \cap \bar{p}iR & & \\
\mathbf{r} = \mathbf{s}\bar{\mathbf{s}} \in \mathbb{Q}(\zeta + \zeta^{-1}) \arrow[ur, "\text{Gentry–Szydlo}"] \arrow[ddrr, "\text{abs. norm}"'] & & & \\
 & & & p = \pi \cdot \bar{\pi} \in \mathbb{Q}(i) \arrow[uull, "\text{lift}"'] \\
 & & N(\mathbf{r}) = p \in \mathbb{Q} \arrow[ur, "\text{split}"'] &
\end{tikzcd}$$

Completing the attack

- Previous slide: works if absolute norm is prime
- Generalizes easily if we can factor the absolute norm
- Then: recover a few candidates for $\mathbf{s}_1$, up to a root of unity
- Easy to check correctness
  - compute the corresponding $\mathbf{s}_2$ as $\mathbf{a} \cdot \mathbf{s}_1 \bmod q$
  - should have coefficients in $\{-1, 0, 1\}$ and be sparse
- Multiplying a correct key $(\mathbf{s}_1, \mathbf{s}_2)$ by a root of unity results in an equivalent key, so we are done!
- Attack works for weak keys with factorable absolute norm of $\mathbf{s}_1$ or $\mathbf{s}_2$
  - e.g. norm of the form $N_0 p$ with p prime and N_0 smooth

Completing the attack

- Previous slide: works if absolute norm is prime
- Generalizes easily if we can factor the absolute norm
- Then: recover a few candidates for $\mathbf{s}_1$, up to a root of unity
- Easy to check correctness
  - compute the corresponding $\mathbf{s}_2$ as $\mathbf{a} \cdot \mathbf{s}_1 \bmod q$
  - should have coefficients in $\{-1, 0, 1\}$ and be sparse
- Multiplying a correct key $(\mathbf{s}_1, \mathbf{s}_2)$ by a root of unity results in an equivalent key, so we are done!
- Attack works for weak keys with factorable absolute norm of $\mathbf{s}_1$ or $\mathbf{s}_2$
  - e.g. norm of the form $N_0 p$ with p prime and N_0 smooth

Completing the attack

- Previous slide: works if absolute norm is prime
- Generalizes easily if we can factor the absolute norm
- Then: recover a few candidates for $\mathbf{s}_1$, up to a root of unity
- Easy to check correctness
  - compute the corresponding $\mathbf{s}_2$ as $\mathbf{a} \cdot \mathbf{s}_1 \bmod q$
  - should have coefficients in $\{-1, 0, 1\}$ and be sparse
- Multiplying a correct key $(\mathbf{s}_1, \mathbf{s}_2)$ by a root of unity results in an equivalent key, so we are done!
- Attack works for weak keys with factorable absolute norm of $\mathbf{s}_1$ or $\mathbf{s}_2$
  - e.g. norm of the form $N_0 p$ with p prime and N_0 smooth

Completing the attack

- Previous slide: works if absolute norm is prime
- Generalizes easily if we can factor the absolute norm
- Then: recover a few candidates for $\mathbf{s}_1$, up to a root of unity
- Easy to check correctness
  - compute the corresponding $\mathbf{s}_2$ as $\mathbf{a} \cdot \mathbf{s}_1 \bmod q$
  - should have coefficients in $\{-1, 0, 1\}$ and be sparse
- Multiplying a correct key $(\mathbf{s}_1, \mathbf{s}_2)$ by a root of unity results in an equivalent key, so we are done!
- Attack works for weak keys with factorable absolute norm of $\mathbf{s}_1$ or $\mathbf{s}_2$
  - e.g. norm of the form $N_0 p$ with p prime and N_0 smooth

Completing the attack

- Previous slide: works if absolute norm is prime
- Generalizes easily if we can factor the absolute norm
- Then: recover a few candidates for $\mathbf{s}_1$, up to a root of unity
- Easy to check correctness
  - compute the corresponding $\mathbf{s}_2$ as $\mathbf{a} \cdot \mathbf{s}_1 \bmod q$
  - should have coefficients in $\{-1, 0, 1\}$ and be sparse
- Multiplying a correct key $(\mathbf{s}_1, \mathbf{s}_2)$ by a root of unity results in an equivalent key, so we are done!
- Attack works for weak keys with factorable absolute norm of $\mathbf{s}_1$ or $\mathbf{s}_2$
  - e.g. norm of the form $N_0 p$ with p prime and N_0 smooth

Completing the attack

- Previous slide: works if absolute norm is prime
- Generalizes easily if we can factor the absolute norm
- Then: recover a few candidates for $\mathbf{s}_1$, up to a root of unity
- Easy to check correctness
    - compute the corresponding $\mathbf{s}_2$ as $\mathbf{a} \cdot \mathbf{s}_1 \bmod q$
    - should have coefficients in $\{-1, 0, 1\}$ and be sparse
- Multiplying a correct key $(\mathbf{s}_1, \mathbf{s}_2)$ by a root of unity results in an equivalent key, so we are done!
- Attack works for weak keys with factorable absolute norm of $\mathbf{s}_1$ or $\mathbf{s}_2$
    - e.g. norm of the form $N_0 p$ with p prime and N_0 smooth

Efficiency of the attack

	n	$B = 5$	$B = 65537$	$B = 655373$	$B = 6553733$
BLISS-0	256	3%	3.8%	6%	6.5%
BLISS-I/II	512	1.5%	2%	2.8%	3.7%
BLISS-III/IV	512	1%	1.75%	2%	2.5%

Experimental density of keys with semi-smooth absolute norm ($N = N_0 \cdot p$ with B-smooth N_0) for various BLISS parameters

Field size n	32	64	128	256	512
CPU time	0.6 s	13 s	21 min.	17h 22 min.	38 days
Clock cycles	$\approx 2^{30}$	$\approx 2^{35}$	$\approx 2^{41}$	$\approx 2^{47}$	$\approx 2^{53}$

Average running time of the attack for various field sizes n

Conclusion and countermeasures countermeasures

- Important to investigate implementation attacks on lattice schemes
- Physical attack resistance should be part of the design goals for practical schemes
- We described faults and SCA against BLISS signatures
- Possible countermeasures?
- Against faults:
  - check that the result has $> (1 - \varepsilon) \cdot n$ non zero coeffs.
  - randomize the order of generation of the coefficients? (still risky)
  - use double loop counters!
- Against SCA:
  - compute rejection probability with floating point arithmetic (slow)
  - use a constant-time Bernoulli sampling (doable)
  - prefer a scheme with simpler structure (GLP) and use masking

Conclusion and countermeasures countermeasures

- Important to investigate implementation attacks on lattice schemes
- Physical attack resistance should be part of the design goals for practical schemes
- We described faults and SCA against BLISS signatures
- Possible countermeasures?
- Against faults:
  - check that the result has $> (1 - \varepsilon) \cdot n$ non zero coeffs.
  - randomize the order of generation of the coefficients? (still risky)
  - use double loop counters!
- Against SCA:
  - compute rejection probability with floating point arithmetic (slow)
  - use a constant-time Bernoulli sampling (doable)
  - prefer a scheme with simpler structure (GLP) and use masking

Conclusion and countermeasures countermeasures

- Important to investigate implementation attacks on lattice schemes
- Physical attack resistance should be part of the design goals for practical schemes
- We described faults and SCA against BLISS signatures
- Possible countermeasures?
- Against faults:
  - check that the result has $> (1 - \varepsilon) \cdot n$ non zero coeffs.
  - randomize the order of generation of the coefficients? (still risky)
  - use double loop counters!
- Against SCA:
  - compute rejection probability with floating point arithmetic (slow)
  - use a constant-time Bernoulli sampling (doable)
  - prefer a scheme with simpler structure (GLP) and use masking

Conclusion and countermeasures countermeasures

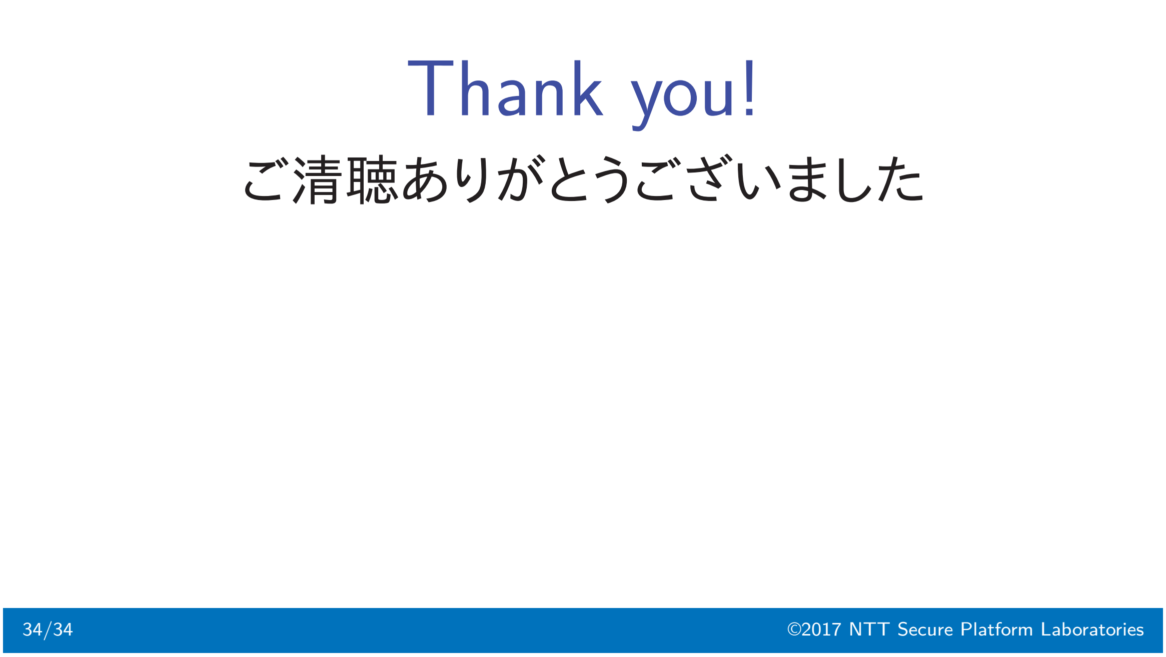
- Important to investigate implementation attacks on lattice schemes
- Physical attack resistance should be part of the design goals for practical schemes
- We described faults and SCA against BLISS signatures
- Possible countermeasures?
- Against faults:
  - check that the result has $> (1 - \varepsilon) \cdot n$ non zero coeffs.
  - randomize the order of generation of the coefficients? (still risky)
  - use double loop counters!
- Against SCA:
  - compute rejection probability with floating point arithmetic (slow)
  - use a constant-time Bernoulli sampling (doable)
  - prefer a scheme with simpler structure (GLP) and use masking

Conclusion and countermeasures countermeasures

- Important to investigate implementation attacks on lattice schemes
- Physical attack resistance should be part of the design goals for practical schemes
- We described faults and SCA against BLISS signatures
- Possible countermeasures?
- Against faults:
  - check that the result has $> (1 - \varepsilon) \cdot n$ non zero coeffs.
  - randomize the order of generation of the coefficients? (still risky)
  - use double loop counters!
- Against SCA:
  - compute rejection probability with floating point arithmetic (slow)
  - use a constant-time Bernoulli sampling (doable)
  - prefer a scheme with simpler structure (GLP) and use masking

Conclusion and countermeasures countermeasures

- Important to investigate implementation attacks on lattice schemes
- Physical attack resistance should be part of the design goals for practical schemes
- We described faults and SCA against BLISS signatures
- Possible countermeasures?
- Against faults:
  - check that the result has $> (1 - \varepsilon) \cdot n$ non zero coeffs.
  - randomize the order of generation of the coefficients? (still risky)
  - use double loop counters!
- Against SCA:
  - compute rejection probability with floating point arithmetic (slow)
  - use a constant-time Bernoulli sampling (doable)
  - prefer a scheme with simpler structure (GLP) and use masking



Thank you!

ご清聴ありがとうございました

Talk at IMI

February 5–7, 2018

Use of algebraic subfield structure in cryptanalysis

Lattice Reductions over Euclidean Rings
with Applications to Cryptanalysis

Taechan Kim
jointly with Changmin Lee at SNU
NTT Secure Platform Laboratories

Backgrounds

Lattice-based cryptography

- post-quantum cryptography
- fully homomorphic encryption
- many other applications

Lattices (classical)

- M: a free $\mathbb{Z}$-module, i.e. closed under addition/multiplication by $\mathbb{Z}$
- A free $\mathbb{Z}$-module has a $\mathbb{Z}$-basis $(b_1, \dots, b_n) \subset M^n$ s.t.

$$M = \mathbb{Z}b_1 \oplus \cdots \oplus \mathbb{Z}b_n$$

- M has “infinitely” many number of basis
- n, the size of the basis, is “invariant” under the choice of the basis.
- n: the rank (or dimension) of M, written as $n := \mathrm{rk}(M)$

Backgrounds

Lattice reduction

- Some basis of lattices are good, but some are not.
- Many lattice problems (SVP/CVP) are easier to solve, if a "good" basis is given.
- Informally, a good basis consists of "reasonably small" and "almost orthogonal" components.
- "Lattice reduction" is to find such a "good" basis from an arbitrary basis.
- LLL-algorithm is one of the most popular algorithms for lattice reduction.
- A key tool not only for lattice-based crypto, but also for various cryptanalysis (e.g. RSA attack)

LLL-algorithm

LLL-algorithm (classical)

- Given a $\mathbb{Z}$-basis of a lattice M, find an LLL-reduced basis.
- (Gram-Schmidt orthogonalization) Given a basis $(b_1, \dots, b_n) \subset \mathbb{Q}^n$, $(b_1^*, \dots, b_n^*) \subset \mathbb{R}^m$ is GS orthogonalization, if $b_1^* = b_1$ and

$$b_i^* = b_i - \sum_{j=1}^{i-1} \mu_{i,j} b_j^* \text{ for } \mu_{i,j} = \langle b_i, b_j^* \rangle / \langle b_j^*, b_j^* \rangle.$$

- (LLL-reduced basis) A basis $(b_1, \dots, b_n)$ is LLL-reduced w.r.t $\delta > 0$ if

$$|\mu_{i,j}| \leq 1/2 \text{ for } 1 \leq j < i \leq n \text{ (size reduced) },$$

$$||b_i^*||^2 \geq (\delta - \mu_{i,i-1}^2)||b_{i-1}^*|| \text{ for } 1 < i \leq n \text{ (Lovasz conditions)}$$

Note

- LLL algorithm outputs a small vector of lattices
  - If $(b_1, \dots, b_n)$ is LLL-reduced w.r.t. $\delta = 3/4$, then $||b_1|| \leq 2^{(n-1)/4} \det(M)^{1/n}$.
- Its running time depends on the size of the dimension, n.

Notations

Number fields

- $h \in \mathbb{Z}[t]$: an irreducible polynomial of degree n.
- $L := \mathbb{Q}[t]/h(t)$: a number field of degree n.
- $\mathbb{Z}_L$: a ring of integers of L (e.g. $\mathbb{Z}_{\mathbb{Q}} = \mathbb{Z}$)

Ideal lattices

- An ideal $\mathcal{I} \subset L$ is a $\mathbb{Z}_L$-submodule of L, thus trivially a $\mathbb{Z}$-module.
- $\mathcal{I}$ is free, since $\mathbb{Z}$ is a PID (Principal Ideal Domain).
- Thus, it is a $\mathbb{Z}$-lattice and called as "ideal-lattice".
- Typically, $\mathrm{rk}(\mathcal{I}) = n = [L : \mathbb{Q}]$.

Motivations

In cryptography,

- L typically has a proper subfield $K \neq \mathbb{Q}$.
- E.g. ℓ-th cyclotomic field $L = \mathbb{Q}[t]/(t^{\ell}+1)$ has a subfield $K = \mathbb{Q}[t]/(t^{k}+1)$, where $\ell = 2^{l}$ for some $l > 0$ and $k \mid \ell$.

Ideal lattice as a $\mathbb{Z}_K$-module

- An ideal $\mathcal{I} \subset L$ is also a $\mathbb{Z}_K$-module for a subfield $K \subset L$.
- If $\mathbb{Z}_K$ is a PID, then $\mathcal{I}$ is a free $\mathbb{Z}_K$-module ($\mathbb{Z}_K$-lattice), thus

$$\mathcal{I} = \mathbb{Z}_K\beta_1 \oplus \cdots \oplus \mathbb{Z}_K\beta_d$$

for a basis $(\beta_1, \ldots, \beta_d) \in L^d$ and $d = [L : K]$.

Matrices of Lattices

- We consider a $\mathbb{Z}_K$-lattice as a $d \times d$ matrix $M_{\mathcal{I}} = [\beta_1|\ldots|\beta_d] \in K^{d\times d}$, instead of $[b_1|\ldots|b_n] \in \mathbb{Q}^{n\times n}$, where $n = [L : \mathbb{Q}]$ and $d = [L : K]$.

Our goal

Motivation

- If a $\mathbb{Z}$-lattice $M \subset L$ can also be considered as a free $\mathbb{Z}_K$-module, $rk_{\mathbb{Z}_K}(M) = [L : K] = d$ is smaller than $rk_{\mathbb{Z}}(M) = [L : \mathbb{Q}] = n$.
- Smaller dimension, faster LLL-algorithm?

LLL algorithm over $\mathbb{Z}_K$

- We restrict our concern to "norm-Euclidean domain" $\mathbb{Z}_K$ (that is, Euclidean domain w.r.t. algebraic norm $N_{K/\mathbb{Q}}$).
- We propose two heuristic LLL algorithms running over $\mathbb{Z}_K$-lattices.

Technical hurdles

- For "GS orthogonalization", one needs to define "inner product" over $K^d \times K^d$.
- What would be analogous notions for "size reduced" and "Lovasz conditions"?

Related works

Related works

- (Napias '96) over Gaussian integers, more generally, quadratic norm-Euclidean domain
  - use "Hermitian product";
  - Euclidean norm (induced by Hermitian product) and algebraic norm coincides, i.e. $N_{K/\mathbb{Q}}(a + b\iota) = a^2 + b^2 = ||(a, b)||^2$.
- (Fieker-Phost '96) over arbitrary Dedekind domain, using pseudo-basis
  - inner product induced by Hermitian product;
  - in a size-reduction step, given $a \in K$, tried to find $q \in \mathbb{Z}_K$ s.t. $Tr_{K/\mathbb{Q}}((a - q)\overline{(a - q)})$ is minimal (in general, not easy to do so).
- (Gan-Ling-Mow '09) over complex fields
  - basically the same as Napias's;
  - does not consider the number field structures.
- (Fieker-Stehlé '10) over arbitrary Dedekind domain, using pseudo-basis
  - convert $\mathbb{Z}_K$-lattice into $\mathbb{Z}$-lattice of a higher dimension;
  - LLL-algorithm is carried over $\mathbb{Z}$-lattice.

Mathematical Background

Euclidean domain

- A ring R is Euclidean if $\exists \phi : R \to \mathbb{N}$ s.t. $\phi(a) \leq \phi(ab)$ for $0 \neq a, b \in R$ and there exists q and $r \in R$ s.t.

$$a = bq + r \text{ with } r = 0 \text{ or } \phi(r) < \phi(b).$$

- E.g. $\mathbb{Z}$ with $\phi(a) = |a|$ and division algorithm

Norm-Euclidean domain

- If $\mathbb{Z}_K$ in K is Euclidean w.r.t. $\phi(a) = |N_{K/\mathbb{Q}}(a)|$, then $\mathbb{Z}_K$ is called norm-Euclidean.
- (Example 1.) $\mathbb{Z}_K$ for $K = \mathbb{Q}(\zeta_k)$, the k-th cyclotomic field, is norm-Euclidean iff

$$k \in \{1, 3, 4, 5, 7, 8, 9, 11, 12, 15, 16, 20, 24\}^{a}$$

- (Example 2.) $\mathbb{Z}_K$ for $K = \mathbb{Q}(\sqrt{-\alpha}, \sqrt{\beta})$ is norm-Euclidean iff

$$\begin{aligned} \alpha &= 1, \quad \beta = 2, 3, 5, 7; \\ \alpha &= 2, \quad \beta = -3, 5; \\ \alpha &= 3, \quad \beta = 2, 5, -7, -11, 17, -19; \\ \alpha &= 7, \quad \beta = 5.^{b} \end{aligned}$$

[a] [Lenstra '75, Masley '75, Ojala '79]

[b] [Lemmermeyer '11]

Mathematical Background

Euclidean minimum

- "norm-Euclideanity" leads us to consider Euclidean minimum of K, $\mathfrak{M}(K)$.
- For any $\xi \in K$, $\exists q \in \mathbb{Z}_K$ s.t. $|N_{K/\mathbb{Q}}(\xi - q)| < \mathfrak{M}(K) < 1$.
- E.g. $\mathfrak{M}(\mathbb{Q}) = 1/2$

Euclidean minimum of $K = \mathbb{Q}(\zeta_k)$

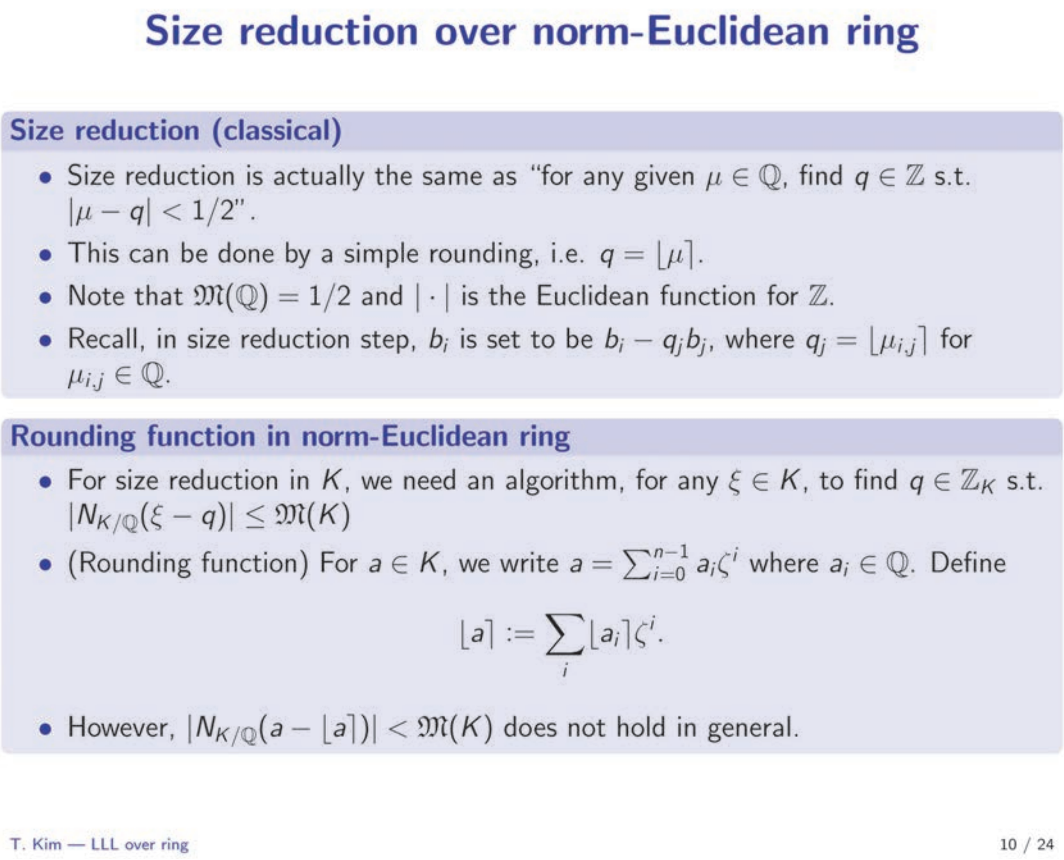
k	1	3	4	5	7	8	9	12	15	16	20	24
$\mathfrak{M}(\mathbb{Q}(\zeta_k))$	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{1}{2}$	$\frac{1}{5}$	$\frac{1}{7}$	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{1}{4}$	$\frac{1}{16}$	$\frac{1}{2}$	$\frac{1}{5}$	$\frac{1}{4}$

Table: Euclidean minimum of k-th cyclotomic fields [Lezowski '14]

Euclidean minimum of $K = \mathbb{Q}(\sqrt{-\alpha}, \sqrt{\beta})$

(α, β)	(1,2)	(1,3)	(1,5)	(1,7)	(2,-3)	(2,5)	(3,2)
$\mathfrak{M}(\mathbb{Q}(\sqrt{-\alpha}, \sqrt{\beta}))$	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{5}{16}$	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{11}{16}$	$\geq \frac{1}{4}$
(α, β)	(3,5)	(3,-7)	(3,-11)	(3,17)	(3,-19)	(7,5)	
$\mathfrak{M}(\mathbb{Q}(\sqrt{-\alpha}, \sqrt{\beta}))$	$\frac{1}{4}$	$\frac{4}{9}$	≤ 0.46	$\frac{13}{16}$	< 0.95	$\frac{9}{16}$	

Table: Euclidean minimum of biquadratic fields [Lemmermeyer '11]



Size reduction over norm-Euclidean ring

Size reduction (classical)

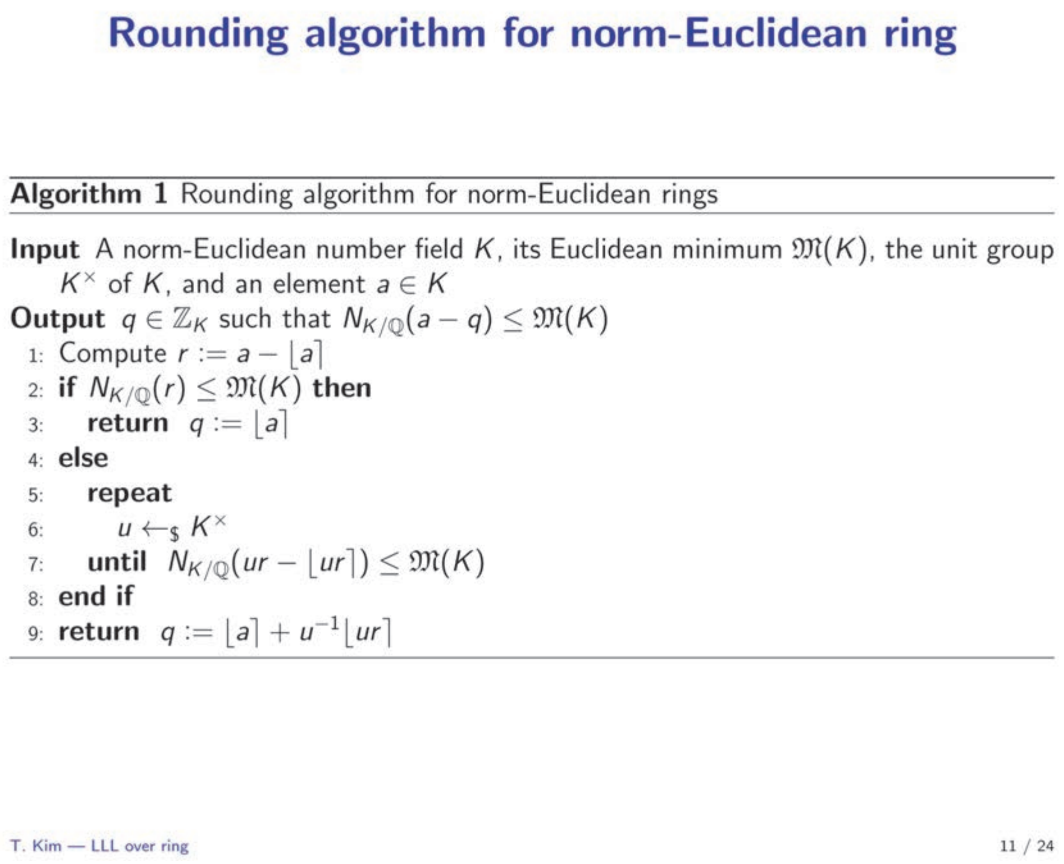
- Size reduction is actually the same as "for any given $\mu \in \mathbb{Q}$, find $q \in \mathbb{Z}$ s.t. $|\mu - q| < 1/2$".
- This can be done by a simple rounding, i.e. $q = \lfloor \mu \rceil$.
- Note that $\mathfrak{M}(\mathbb{Q}) = 1/2$ and $|\cdot|$ is the Euclidean function for $\mathbb{Z}$.
- Recall, in size reduction step, b_i is set to be $b_i - q_j b_j$, where $q_j = \lfloor \mu_{i,j} \rceil$ for $\mu_{i,j} \in \mathbb{Q}$.

Rounding function in norm-Euclidean ring

- For size reduction in K, we need an algorithm, for any $\xi \in K$, to find $q \in \mathbb{Z}_K$ s.t. $|N_{K/\mathbb{Q}}(\xi - q)| \leq \mathfrak{M}(K)$
- (Rounding function) For $a \in K$, we write $a = \sum_{i=0}^{n-1} a_i \zeta^i$ where $a_i \in \mathbb{Q}$. Define

$$\lfloor a \rceil := \sum_i \lfloor a_i \rceil \zeta^i.$$

- However, $|N_{K/\mathbb{Q}}(a - \lfloor a \rceil)| < \mathfrak{M}(K)$ does not hold in general.



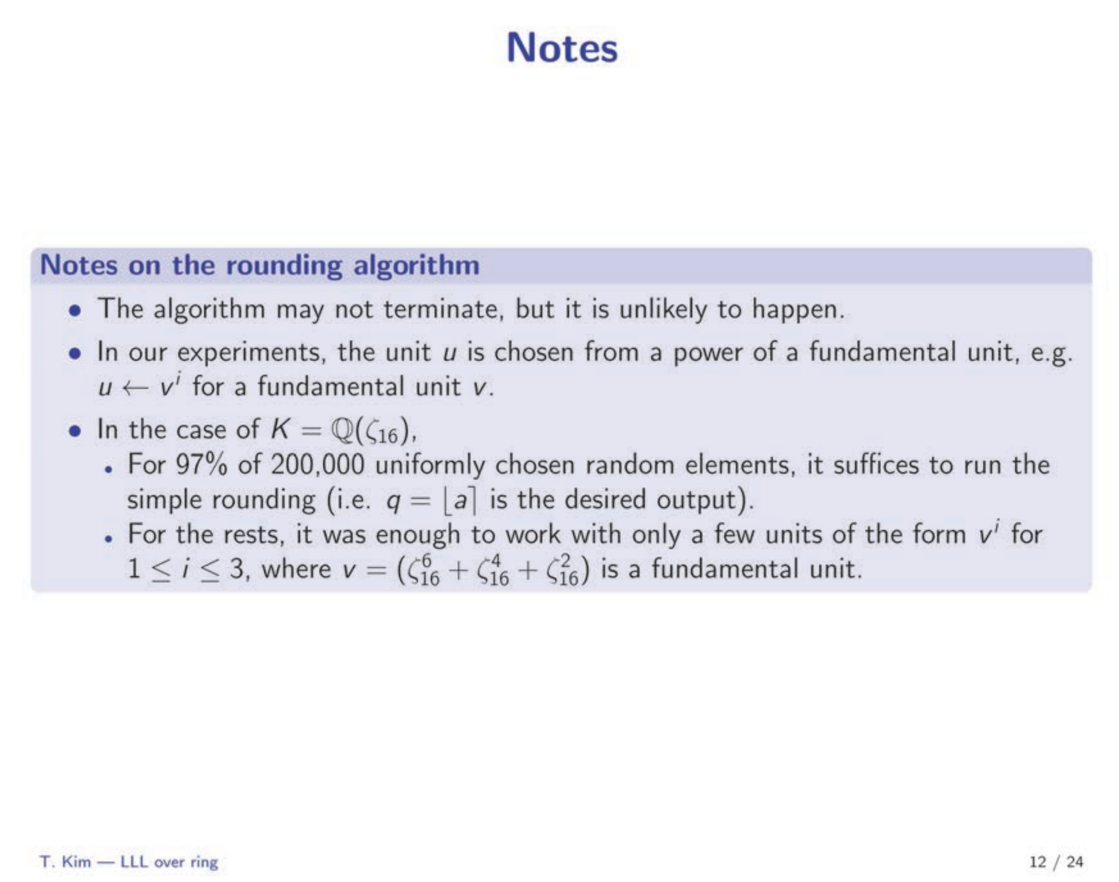
Rounding algorithm for norm-Euclidean ring

Algorithm 1 Rounding algorithm for norm-Euclidean rings

Input A norm-Euclidean number field K, its Euclidean minimum $\mathfrak{M}(K)$, the unit group $K^\times$ of K, and an element $a \in K$

Output $q \in \mathbb{Z}_K$ such that $N_{K/\mathbb{Q}}(a - q) \leq \mathfrak{M}(K)$

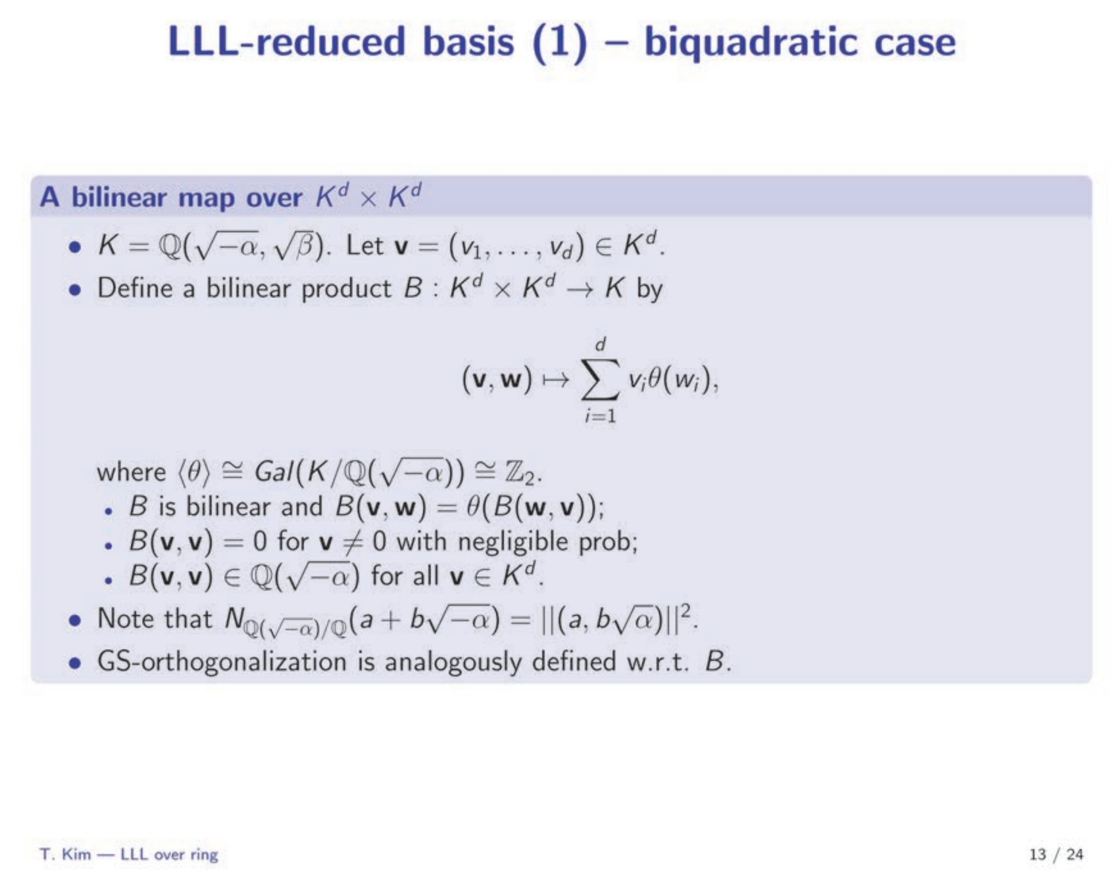
```
Compute $r := a - \lfloor a \rceil$
if $N_{K/\mathbb{Q}}(r) \leq \mathfrak{M}(K)$ then
    return $q := \lfloor a \rceil$
else
    repeat
        $u \leftarrow_\$ K^\times$
    until $N_{K/\mathbb{Q}}(ur - \lfloor ur \rceil) \leq \mathfrak{M}(K)$
end if
return $q := \lfloor a \rceil + u^{-1} \lfloor ur \rceil$
```



Notes

Notes on the rounding algorithm

- The algorithm may not terminate, but it is unlikely to happen.
- In our experiments, the unit u is chosen from a power of a fundamental unit, e.g. $u \leftarrow v^i$ for a fundamental unit v.
- In the case of $K = \mathbb{Q}(\zeta_{16})$,
  - For 97% of 200,000 uniformly chosen random elements, it suffices to run the simple rounding (i.e. $q = \lfloor a \rceil$ is the desired output).
  - For the rests, it was enough to work with only a few units of the form v^i for $1 \leq i \leq 3$, where $v = (\zeta_{16}^6 + \zeta_{16}^4 + \zeta_{16}^2)$ is a fundamental unit.



LLL-reduced basis (1) – biquadratic case

A bilinear map over $K^d \times K^d$

- $K = \mathbb{Q}(\sqrt{-\alpha}, \sqrt{\beta})$. Let $\mathbf{v} = (v_1, \ldots, v_d) \in K^d$.
- Define a bilinear product $B : K^d \times K^d \to K$ by

$$(\mathbf{v}, \mathbf{w}) \mapsto \sum_{i=1}^{d} v_i \theta(w_i),$$

  where $\langle \theta \rangle \cong Gal(K/\mathbb{Q}(\sqrt{-\alpha})) \cong \mathbb{Z}_2$.
  - B is bilinear and $B(\mathbf{v}, \mathbf{w}) = \theta(B(\mathbf{w}, \mathbf{v}))$;
  - $B(\mathbf{v}, \mathbf{v}) = 0$ for $\mathbf{v} \neq 0$ with negligible prob;
  - $B(\mathbf{v}, \mathbf{v}) \in \mathbb{Q}(\sqrt{-\alpha})$ for all $\mathbf{v} \in K^d$.
- Note that $N_{\mathbb{Q}(\sqrt{-\alpha})/\mathbb{Q}}(a + b\sqrt{-\alpha}) = ||(a, b\sqrt{\alpha})||^2$.
- GS-orthogonalization is analogously defined w.r.t. B.

LLL-reduced basis (1) – biquadratic case

LLL-reduced condition

- A basis $(\beta_1, \ldots, \beta_d) \subset K^d$ is called $\mathbb{Z}_K$-LLL-reduced w.r.t $\delta > 0$ if
  1. $N_{K/\mathbb{Q}}(\mu_{i,j}) \leq \mathfrak{M}(K)$ for $1 \leq j < i \leq n$ (size reduced)
  2. $||B_i|| \geq (\delta - ||\mu_{i,i-1}\theta(\mu_{i,i-1})||) \cdot ||B_{i-1}||$ for $1 < i \leq n$ (Lovasz condition), where $B_i = B(\beta_i^*, \beta_i^*) \in \mathbb{Q}(\sqrt{-\alpha})$.

Proposition

- Let $(\beta_1, \ldots, \beta_d)$ be $\mathbb{Z}_K$-LLL-reduced basis of L w.r.t. δ, then

$$||B_1|| \leq (\delta - \mathfrak{M}(K)^{1/2})^{-(d-1)/2} N_{K/\mathbb{Q}}(\det(L))^{1/d}.$$

LLL-algorithm over biquadratic field

Input a basis $\{\mathbf{b}_1, \cdots, \mathbf{b}_d\}$ of $M \subset \mathbb{Z}_K^d$, $\mathfrak{M}(K)$, the unit group $K^\times$, and $\delta > 0$.

Output $\mathbb{Z}_K$-LLL-reduced basis $\{\mathbf{b}_1, \cdots, \mathbf{b}_d\}$.

```
Compute the Gram-Schmidt basis $\{\mathbf{b}_1^*, \cdots, \mathbf{b}_d^*\}$ with respect to the bilinear map $B(\cdot,\cdot)$
Compute the coefficients $\mu_{i,j} = B(\mathbf{b}_i, \mathbf{b}_j^*)/B(\mathbf{b}_j^*, \mathbf{b}_j^*)$ for $1 \leq j < i \leq d$ and $B_i = B(\mathbf{b}_i^*, \mathbf{b}_i^*)$ for $1 \leq i \leq d$.
Set $k = 2$
while $k \leq d$ do
    for $j = k - 1$ to 1 do
        Compute $q_j \in \mathbb{Z}_K$ such that $N_{K/\mathbb{Q}}(\mu_{k,j} - q_j) \leq \mathfrak{M}(K)$ using Algorithm 1
        Set $\mathbf{b}_k = \mathbf{b}_k - q_j \cdot \mathbf{b}_j$
        Update $\mu_{k,j} = B(\mathbf{b}_k, \mathbf{b}_j^*)/B(\mathbf{b}_j^*, \mathbf{b}_j^*)$ and $B_k$ for $1 \leq j \leq k$
    end for
    if $\|B_k\| \geq \left(\delta - N_{K/\mathbb{Q}}(\mu_{k,k-1})^{1/2}\right) \cdot \|B_{k-1}\|$ then
        $k = k + 1$
    else
        Swap $\mathbf{b}_k$ and $\mathbf{b}_{k-1}$
        Update $\mathbf{b}_k^*, \mathbf{b}_{k-1}^*, B_k, B_{k-1}$, and $\mu_{i,j}$ for $1 \leq i, j \leq s$
        $k = \min\{2, k - 1\}$
    end if
end while
```

LLL-reduced basis (2) – general case

Hermitian product

- Define a blinear map $H : K^d \times K^d \to K$ by

$$(\mathbf{v}, \mathbf{w}) \mapsto \sum_{i=1}^{d} v_i \overline{w}_i,$$

  where $\bar{\cdot}$ denotes the complex conjugation.
- GS-orthogonalization is analogously defined w.r.t. H.

LLL-reduced condition

- A basis $(\beta_1, \ldots, \beta_d) \subset K^d$ is called $\mathbb{Z}_K$-LLL-reduced w.r.t $\delta > 0$ if
  - $N_{K/\mathbb{Q}}(\mu_{i,j}) \leq \mathfrak{M}(K)$ for $1 \leq j < i \leq n$ (size reduced)
  - $N_{K/\mathbb{Q}}(B_i + \mu_{i,i-1}\overline{\mu_{i,i-1}}B_{i-1}) \geq \delta \cdot N_{K/\mathbb{Q}}(B_{i-1})$ for $1 < i \leq n$ (Lovasz condition), where $B_i = H(\beta_i^*, \beta_i^*)$.

Cautions

- Unlike the classical case, the Lovasz condition cannot be replaced with $N_{K/\mathbb{Q}}(B_i) \geq (\delta - \mu_{i,i-1}\overline{\mu_{i,i-1}}) \cdot N_{K/\mathbb{Q}}(B_{i-1})$.
- This is because the triangle inequality does not hold w.r.t. $N_{K/\mathbb{Q}}$.

LLL-algorithm over norm-Euclidean rings

```
Input  a basis $\{\mathbf{b}_1, \cdots, \mathbf{b}_d\} \subset K^d$, $\mathfrak{M}(K)$, the unit group $K^\times$, and $\delta > 0$.
Output  LLL-reduced basis $\{\mathbf{b}_1, \cdots, \mathbf{b}_d\}$.
Compute the Gram-Schmidt basis $\{\mathbf{b}_1^*, \cdots, \mathbf{b}_d^*\}$ with respect to the bilinear map $H(\cdot,\cdot)$
Compute the coefficients $\mu_{i,j} = H(\mathbf{b}_i, \mathbf{b}_j^*)/H(\mathbf{b}_j^*, \mathbf{b}_j^*)$ for $1 \leq j < i \leq d$ and $B_i = H(\mathbf{b}_i^*, \mathbf{b}_i^*)$ for
  $1 \leq i \leq d$.
Set $k = 2$
while $k \leq d$ do
    for $j = k - 1$to 1  do
        Compute $q_j \in \mathbb{Z}_K$ such that $N_{K/\mathbb{Q}}(\mu_{k,j} - q_j) \leq \mathfrak{M}(K)$ using Algorithm 1
        Set $\mathbf{b}_k = \mathbf{b}_k - q_j \cdot \mathbf{b}_j$
        Update $\mu_{k,j} = H(\mathbf{b}_k, \mathbf{b}_j^*)/H(\mathbf{b}_j^*, \mathbf{b}_j^*)$ and $B_k$ for $1 \leq j \leq k$
    end for
    if $N_{K/\mathbb{Q}}(B_k + \mu_{k,k-1}\overline{\mu_{k,k-1}}B_{k-1}) \geq \delta \cdot N_{K/\mathbb{Q}}(B_{k-1})$ then
        $k = k + 1$
    else
        Swap $\mathbf{b}_k$ and $\mathbf{b}_{k-1}$
        Update $\mathbf{b}_k^*, \mathbf{b}_{k-1}^*, B_k, B_{k-1}$, and $\mu_{i,j}$ for $1 \leq i, j \leq s$
        $k = \min\{2, k - 1\}$
    end if
end while
```

Experimental Results

Lattices

- The lattice $\mathcal{L}$ is generated by rows of the matrix in $K^{d \times d}$,

$$\begin{pmatrix} q & 0 & \cdots & \cdots & 0 \\ \gamma_1 & 1 & \cdots & \cdots & 0 \\ \gamma_2 & 0 & 1 & \ddots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \gamma_{d-1} & 0 & \cdots & \cdots & 1 \end{pmatrix},$$

  where q and $\gamma_i \in \mathbb{Z}_K$.
- This shape of the lattice basis appears in several cryptanalysis.

Parameter choices

- We carried out the reduction of lattices over $K = \mathbb{Q}(\zeta_k)$ for $k = 5, 8, 16$ of dimension $10 \leq d \leq 50$.
- If the lattices are considered as $\mathbb{Z}$-lattices, the dimension 50 over $\mathbb{Z}_K$ corresponds to the dimension 200 over $\mathbb{Z}$ when $k = 5$ or 8, or 400 when $k = 16$.
- $\delta = 3/4$

Experimental Results

Output quality

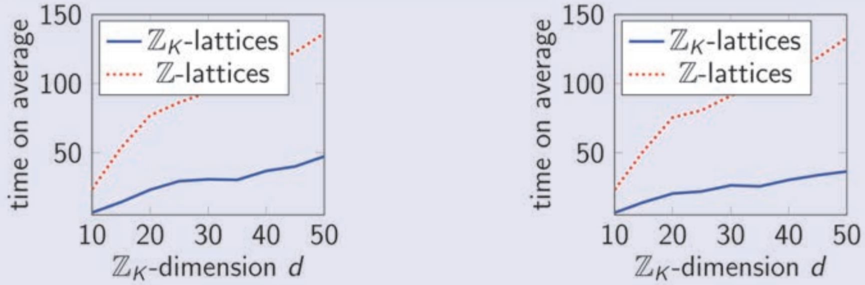
- Theoretically, we have no guarantee on the output quality.
- In biquadratic case, it is possible to guarantee output quality using a different type of inner product, but its practical performance is worse than general case that uses Hermitian product (see our paper).
- Let $n = d \cdot [K : \mathbb{Q}]$, a dimension of $\mathcal{L}$ over $\mathbb{Z}$
- For $\mathbf{v} := (v_1, \dots, v_d) \in K^d$, define $||\mathbf{v}||_\infty = \max_i ||v_i||_\infty$.
- As a measure of output quality, we use

$$C := \frac{||\mathbf{b}_1||_\infty}{N_{K/\mathbb{Q}}(\det(\mathcal{L}))^{1/n}},$$

  where the volume of $\mathcal{L}$ over $\mathbb{Z}$ is the same as $N_{K/\mathbb{Q}}(\det(\mathcal{L})) = N_{K/\mathbb{Q}}(q)$.
- The classical Hermite factor is defined with $|| \cdot ||_2$ norm.
- Heuristically, we observe that $C \approx 1.02^n$.
- Taking $||\mathbf{v}||_2 \leq \sqrt{n}||\mathbf{v}||_\infty$ into account, the Hermite constant of our reduction is $\lesssim 1.02n^{1/2n}$.
- As n grows, Hermite constant becomes close to the average Hermite factor 1.02.

Experimental Results

Timing results



(a) $\mathbb{Z}_K = \mathbb{Z}[\zeta_5]$

(b) $\mathbb{Z}_K = \mathbb{Z}[\zeta_8]$

Figure: Time comparison of running time

Note

- Our $\mathbb{Z}_K$-lattice reduction is about 3 times faster than the classical reduction done over $\mathbb{Z}$.
- We did not attempt to compare our naive implementation with already well-optimized LLL implementation.
- For the consistency of the comparison, we used our own implementation for both $\mathbb{Z}$ and $\mathbb{Z}_K$-reduction (see our SAGE codes in Appendix).

Applications

Sieving in exTNFS (by K.-Barbulescu)

- exTNFS is a best known algorithm to solve the DLP over $\mathbb{F}_{p^n}$ (n: composite, p: not small).
- In a step called special-q method in exTNFS method, we need to consider: find a small basis of the lattice

$$M_{\mathfrak{Q}} := \left\{ (a_0, \ldots, a_{\tau-1}) \in \mathbb{Z}[\iota]^\tau : \left(\sum_{i=0}^{\tau-1} a_i \alpha_f \right) \equiv 0 \mod \mathfrak{Q} \right\},$$

where $\mathfrak{Q}$ is a prime ideal in $\mathbb{Z}_L$ for $L = K[\alpha] = K[x]/f(x)$ and $K = \mathbb{Q}[\iota] = \mathbb{Q}[t]/h(t)$.

- $M_{\mathfrak{Q}}$ is the τ-dimensional $\mathbb{Z}_K$-lattice.
- A classical approach is to consider the lattice as $\mathbb{Z}$-lattice.

Parameters for BN curves

- Set $h(t) = \Phi_5(t) = t^4 + t^3 + t^2 + t + 1$ so that $K = \mathbb{Q}(\zeta_5)$ and $\mathbb{Z}_K = \mathbb{Z}[\zeta_5]$.
- Set $f(x) = x^3 - x^2 - u$, where $p = P(u)$ and $P(x) = 36x^4 + 36x^3 + 24x^2 + 6x + 1$ and $u = 2^{158} - 2^{128} - 2^{68} + 1$
- Set $L = K[x]/f(x)$.

Special-$\mathfrak{q}$ in BN curve parameters

Example (cont.)

- Take a prime ideal $\mathfrak{Q} = \langle \mathfrak{q}, \alpha_f - \gamma \rangle \subset L$ and $\tau = 2$.
- where,

$$\mathfrak{q} = (q) = \big(-461479\zeta_5^3 - 383970\zeta_5^2 - 265505\zeta_5 - 303923 \big);$$

$$\gamma = 16946578643505257763313.$$

- Then $M_{\mathfrak{Q}} = \begin{pmatrix} q & 0 \\ -\gamma & 1 \end{pmatrix}$.
- We obtained

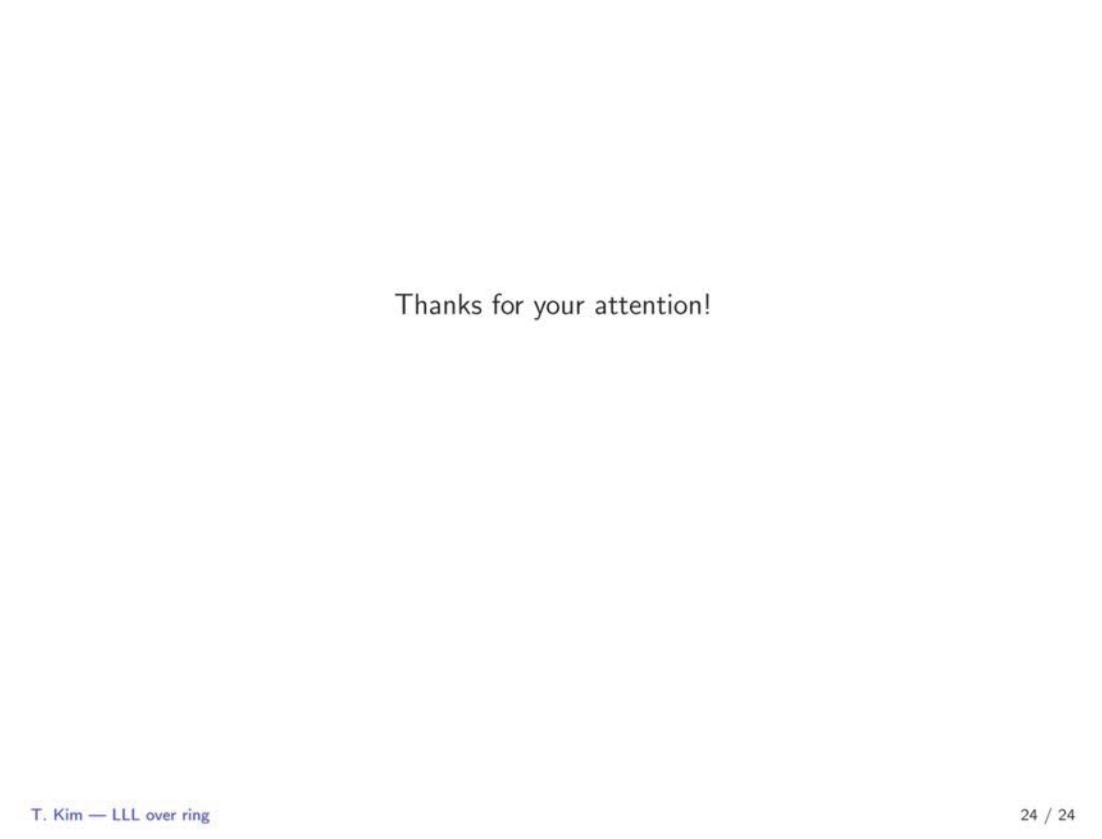
$$\mathrm{LLL}\,(M_{\mathfrak{Q}}) = \begin{pmatrix} 532\zeta_5^3 + 850\zeta_5^2 + 179\zeta_5 - 464 & 224\zeta_5^3 + 132\zeta_5^2 - 13\zeta_5 + 367 \\ -649\zeta_5^3 + 186\zeta_5^2 + 661\zeta_5 + 73 & 11\zeta_5^3 - 264\zeta_5^2 + 35\zeta_5 - 71 \end{pmatrix}.$$

- Note that $\log_2(850) \approx 9.73$ and $\log_2 \left(N_{K/\mathbb{Q}}(q)^{\frac{1}{\tau m}} \right) \approx 9.29$.

Future works

Open questions

- Apply to cryptanalysis of lattice-based cryptography
- BKZ, lattice enumeration over Euclidean ring?
- Prove the output quality



Thanks for your attention!

Fine-grained complexity and cryptography: A personal survey

Suguru TAMAKI *Kyoto University*

Workshop on analysis of mathematical cryptography via algebraic methods
February 5, 2017, Nishijin Plaza, Fukuoka

Talk Plan

1. Fine-grained complexity
2. Fine-grained complexity and cryptography

[Hardness results]

- ➢ Hardness of SVP/CVP
- ➢ Average-case hardness
- ➢ Gap-ETH from ``strong” OWF

[Easiness results]

- ➢ Beating Brute Force for Systems of Polynomial Equations over Finite Fields

References

- [VW18] Virginia Vassilevska Williams: Survey on Fine-Grained Complexity. *ICM 2018*, to appear

- Many papers in *STOC*, *FOCS*, *SODA*, *ICALP* etc.

Complexity Theory

Goal

1. understand the power/limits of computational models: (non)deterministic/probabilistic/quantum Turing machines, Boolean circuits, etc.

2. classify computational problems according to the amount of resources to solve them:
time, space, randomness, etc.

Complexity Theory

The most popular classification criteria
Is Problem X in P or NP-hard?

Easiness give a polynomial time algorithm for X

Hardness show X is NP-hard via a polynomial time reduction from some NP-hard problem

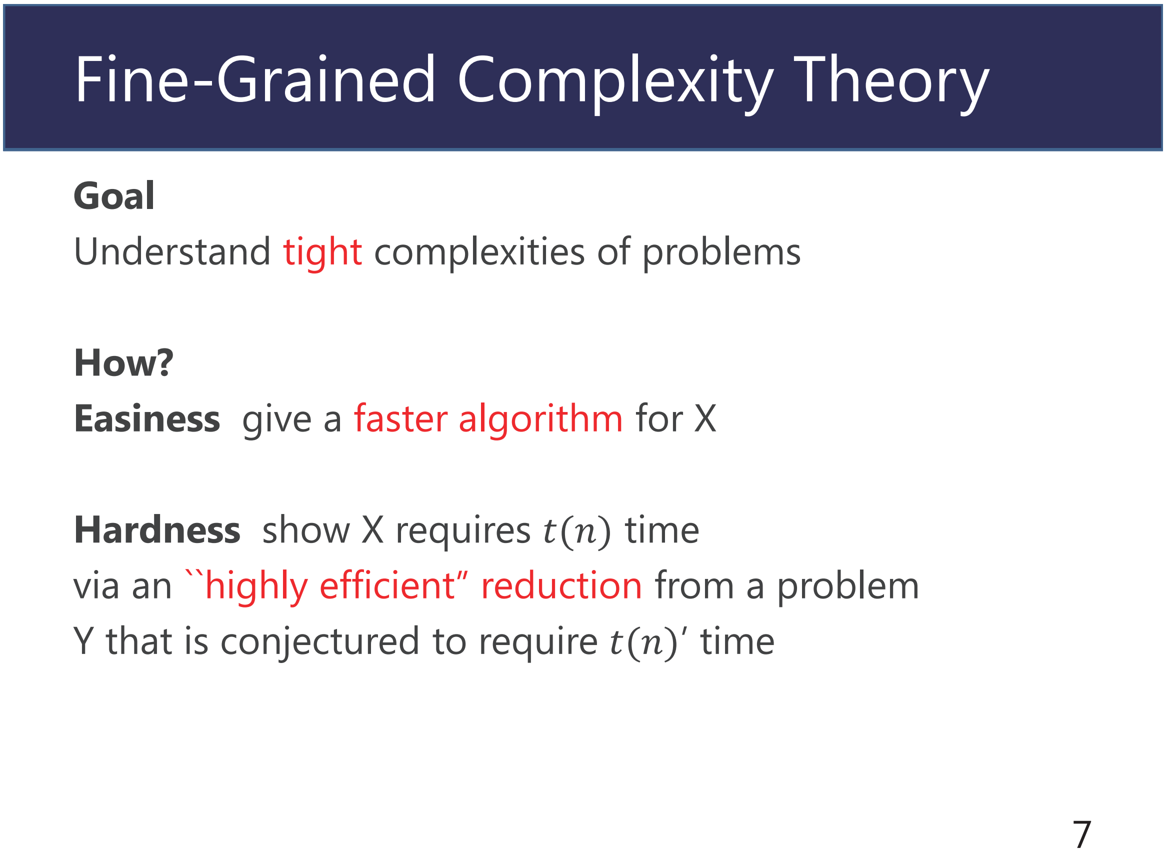
Question Are we satisfied with this classification?

Complexity Theory

Possible complaints about theory of NP-hardness
What is the exact time complexity of Problem X?

Case [X is in P]:
Is X solvable in linear time?
Does X require quadratic time?

Case [X is NP-hard]:
Is X solvable in sub-exponential time?
Does X require exponential time?



Fine-Grained Complexity Theory

Goal

Understand tight complexities of problems

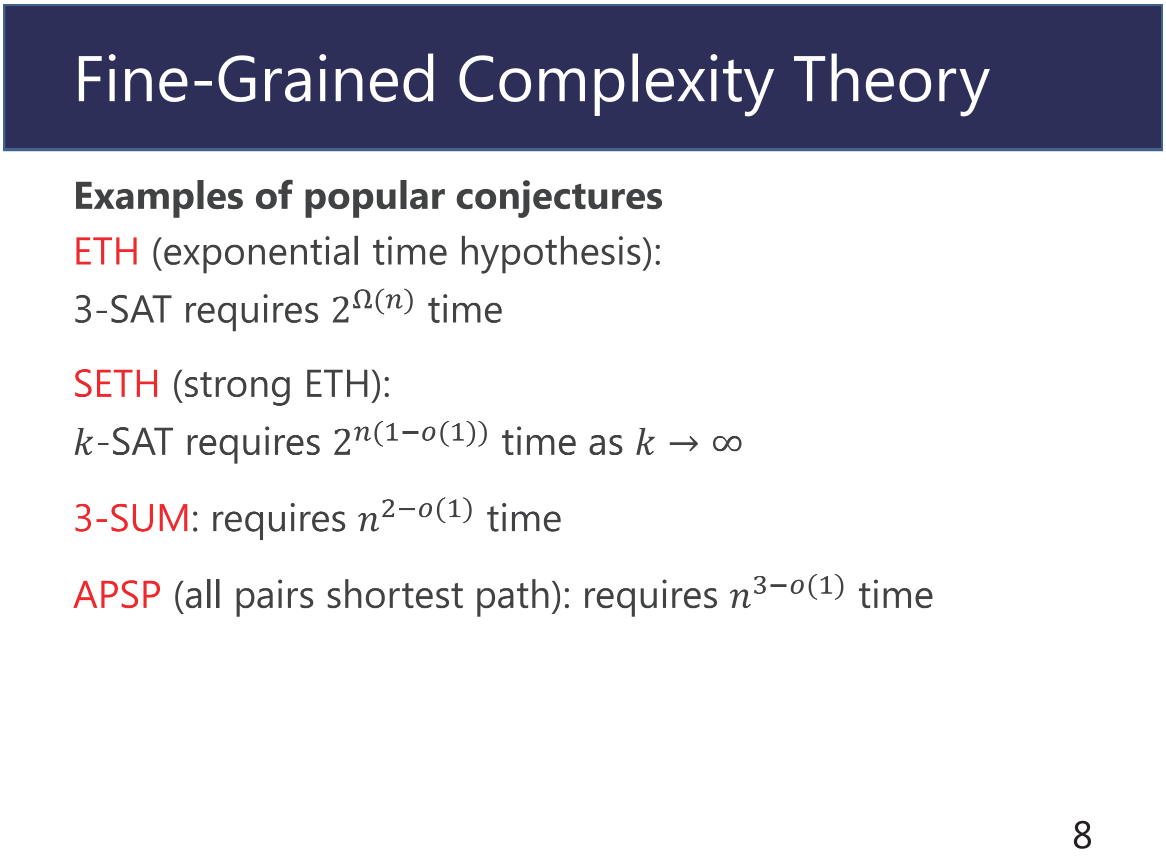
How?

Easiness give a faster algorithm for X

Hardness show X requires $t(n)$ time

via an ``highly efficient” reduction from a problem

Y that is conjectured to require $t(n)'$ time



Fine-Grained Complexity Theory

Examples of popular conjectures

ETH (exponential time hypothesis):

3-SAT requires $2^{\Omega(n)}$ time

SETH (strong ETH):

k-SAT requires $2^{n(1-o(1))}$ time as $k \rightarrow \infty$

3-SUM: requires $n^{2-o(1)}$ time

APSP (all pairs shortest path): requires $n^{3-o(1)}$ time

(S)ETH

literal: Boolean variable or its negation

k-clause: disjunction of at most k literals

k-CNF: conjunction of k-clauses

k-SAT: given a k-CNF, is there an assignment to the variables s.t. all the clauses become true?

$$f(x) = (\neg x_1 \vee x_2 \vee x_3)(x_1 \vee \neg x_3 \vee \neg x_4)(x_2 \vee x_3 \vee x_4) \cdots$$

Upper Bounds: k -SAT is in time $2^{n(1-1/k)}$ [PPZ97,...]

ETH: 3-SAT requires $2^{\Omega(n)}$ time

SETH: k-SAT requires $2^{n(1-o(1))}$ time as $k \to \infty$

Example: SETH hardness of OV

Orthogonal Vector (OV)

Input: $U, V \subseteq \{0,1\}^d, |U| = |V| = n$

Output: $\exists (x, y) \in U \times V$ s.t. $\sum_{i=1}^{d} x_i y_i = 0$?

Trivial upper bound

OV is solvable in time $O(n^2 \mathrm{poly}(\log n))$

Theorem [W05]

SETH implies OV requires $\Omega(n^{2-o(1)})$ time as $d \to \infty$

Reduction from k-SAT to OV

k-SAT instance: CNF of n variable and m clauses

$(\neg x_1 \vee x_2 \vee x_3)(x_1 \vee \neg x_3 \vee \neg x_4)(x_2 \vee x_3 \vee x_4) \cdots$

OV instance: $U = \{u_y\}, V = \{v_y\}$,

$U, V \subseteq \{0,1\}^m, |U| = |V| = 2^{n/2}$

For $y \in \{0,1\}^{n/2}$,

$$u_{y,i} = \begin{cases} 0 & \text{if } (x_1, \ldots, x_{n/2}) = y \text{ makes } i\text{th clause true} \\ 1 & \text{otherwise} \end{cases}$$

$$v_{y,i} = \begin{cases} 0 & \text{if } (x_{n/2+1}, \ldots, x_n) = y \text{ makes } i\text{th clause true} \\ 1 & \text{otherwise} \end{cases}$$

$\sum u_{y,i} v_{y',i} > 0$ if and only if (y, y') makes some clause false

11

Reduction from k-SAT to OV

k-SAT instance: CNF of n variable and m clauses

$(\neg x_1 \vee x_2 \vee x_3)(x_1 \vee \neg x_3 \vee \neg x_4)(x_2 \vee x_3 \vee x_4) \cdots$

OV instance: $U = \{u_y\}, V = \{v_y\}$,

$U, V \subseteq \{0,1\}^m, |U| = |V| = 2^{n/2}$

OV with $|U| = |V| = n$ is solvable in $O(n^{2-\delta})$

implies

k-SAT is solvable in in $O\left(2^{\frac{n}{2}(1-\delta)}\right) = O\left(2^{n(1-\delta/2)}\right)$

12

Talk Plan

1. Fine-grained complexity
2. Fine-grained complexity and cryptography

[Hardness results]

- Hardness of SVP/CVP
- Average-case hardness
- Gap-ETH from ``strong” OWF

[Easiness results]

- Beating Brute Force for Systems of Polynomial Equations over Finite Fields

References

- [BGSD17] Huck Bennett, Alexander Golovnev, Noah Stephens-Davidowitz: On the Quantitative Hardness of CVP. *FOCS 2017*

- [ASD17] Divesh Aggarwal, Noah Stephens-Davidowitz: (Gap/S)ETH Hardness of SVP. arXiv:1712.00942

Shortest/Closest Vector Problems

Lattice of rank n and dimension d

For linearly independent vectors $b_1, \dots, b_n \in \mathbb{R}^d$,
$L(b_1, \dots, b_n) \coloneqq \{\sum_{i=1}^{n} z_i b_i : z_i \in \mathbb{Z}\}$

SVP$_p$ compute the minimum ℓ_p-length of a non-zero vector in $L(b_1, \dots, b_n)$

CVP$_p$ compute the minimum ℓ_p-distance of a vector $t \in \mathbb{R}^d$ and $L(b_1, \dots, b_n)$

15

Motivation

Lattice of rank n and dimension d

For linearly independent vectors $b_1, \dots, b_n \in \mathbb{R}^d$,
$L(b_1, \dots, b_n) \coloneqq \{\sum_{i=1}^{n} z_i b_i : z_i \in \mathbb{Z}\}$

Question

The best exact algorithms for e.g. SVP$_2$ and CVP$_2$
run in time $2^{n(1+o(1))}$ [ADS15]
Can we improve the running time to $2^{o(n)}$ or 1.1^n ?

16

M2SH

Max 2-SAT:
given a 2-CNF and an integer $l > 0$,
determine whether there exists an assignment
that makes at least l clauses true or not

Max 2-SAT Hypothesis:

Max 2-SAT requires requires $2^{\frac{\omega n}{3}(1-o(1))}$ time
$\omega \geq 2$ is the minimum real s.t. multiplying two $n \times n$ matrices can be computed in time $O(n^{\omega})$

Gap-ETH

Gap(γ)-3-SAT:
given a 3-CNF satisfying either
(i) there is an assignment s.t. all the clauses become true, or
(ii) there is no assignment s.t. at least a $(1-\gamma)$ faction of the clauses become true,
decide (i) or (ii)

Gap-ETH: For some $\gamma > 0$, Gap(γ)-3-SAT requires $2^{\Omega(n)}$ time
Non-uniform Gap-ETH: For some $\gamma > 0$, Gap(γ)-3-SAT requires $2^{\Omega(n)}$ size circuits

Hardness of CVP

Theorem [BGSD17]

1. M2SH ⇒ CVP_2 requires $2^{\frac{\omega n}{3}(1-o(1))}$ time

2. ETH ⇒ CVP_2 requires $2^{\Omega(n)}$ time

3. Gap-ETH ⇒ CVP_2 requires $2^{\Omega(n)}$ time even for α-approximation with some constant $\alpha > 1$

Cf. best exact algorithm in time $2^{n(1+o(1))}$ [ADS15]

Remark

[BGSD17] shows other hardness results including the case of $\mathrm{CVP}_p, p \neq 2$

Hardness of SVP

LKNH: the lattice kissing number is $2^{o(n)}$

Theorem [ASD17]

Non-uniform-Gap-ETH & LKNH

⇒ SVP_2 requires $2^{\Omega(n)}$ time even for α-approximation with some constant $\alpha > 1$

Cf. best exact algorithm in time $2^{n(1+o(1))}$ [ADS15]

Remark

[ASD17] shows other hardness results including the case of $\mathrm{SVP}_p, p \neq 2$

Talk Plan

1. Fine-grained complexity
2. Fine-grained complexity and cryptography

[Hardness results]

- Hardness of SVP/CVP
- Average-case hardness
- Gap-ETH from ``strong" OWF

[Easiness results]

- Beating Brute Force for Systems of Polynomial Equations over Finite Fields

References

- [BRSV17] Marshall Ball, Alon Rosen, Manuel Sabin, Prashant Nalini Vasudevan: Average-case fine-grained hardness. *STOC* 2017

- [GR17] Oded Goldreich, Guy N. Rothblum: Worst-case to Average-case reductions for subclasses of P. ECCC TR17-130, 2017

Motivation

So far, we have been concerned with worst-case complexity

For cryptography, it is convenient to have a problem that is solvable in time $t(n)$ but not in time $t(n)^{1-\delta}$ on average

Can we show such results under some (possibly worst-case hardness) assumption?

Variant of OV

$p(n) > n^2$ be the smallest prime, $d(n) := \lceil \log^2 n \rceil$

Define $f\mathrm{OV}_n : \mathrm{F}_p^{n\times d} \times \mathrm{F}_p^{n\times d} \to \mathrm{F}_p$ as

$f\mathrm{OV}_n(U,V) = \sum_{i,j\in[n]} \prod_{k\in[d]} (1 - U_{ik}V_{jk})$

For $U, V \in \{0,1\}^{n\times d}$,

1. $\prod_{k\in[d]}(1 - U_{ik}V_{jk}) = \begin{cases} 1 & \text{if } \sum_{k\in[d]} U_{ik}V_{jk} = 0 \\ 0 & \text{otherwise} \end{cases}$

2. $f\mathrm{OV}_n(U,V) = \#\{(i,j) \in [n]^2 : \sum_{k\in[d]} U_{ik}V_{jk} = 0\}$

$f\mathrm{OV}_n(U,V)$ can be computed in time $O(n^2 \mathrm{poly}(\log n))$

Average-case hardness of $f\mathrm{OV}_n$

$p(n) > n^2$ be the smallest prime, $d(n) := \lceil \log^2 n \rceil$

Define $f\mathrm{OV}_n : \mathrm{F}_p^{n\times d} \times \mathrm{F}_p^{n\times d} \rightarrow \mathrm{F}_p$ as

$f\mathrm{OV}_n(U, V) = \sum_{i,j\in[n]} \prod_{k\in[d]} (1 - U_{ik} V_{jk})$

Theorem [BRSV17]

If $f\mathrm{OV}_n(U, V)$ can be computed in time $O(n^{1+\alpha})$

for a ¾ fraction of inputs $(U, V) \in \mathrm{F}_p^{n\times d} \times \mathrm{F}_p^{n\times d}$,

then OV is solvable in time $O(n^{1+\alpha})$ in the worst case

(and SETH is false)

Talk Plan

1. Fine-grained complexity

2. Fine-grained complexity and cryptography

[Hardness results]

- Hardness of SVP/CVP
- Average-case hardness
- Gap-ETH from ``strong” OWF

[Easiness results]

- Beating Brute Force for Systems of Polynomial Equations over Finite Fields

References

- [A17] Benny Applebaum: Exponentially-Hard gap-CSP and local PRG via Local Hardcore Functions. *FOCS* 2017

Motivation

Gap(γ)-3-SAT:

given a 3-CNF satisfying either

(i) there is an assignment s.t. all the clauses become true, or

(ii) there is no assignment s.t. at least a $(1-\gamma)$ faction of the clauses become true,

decide (i) or (ii)

Gap-ETH: For some $\gamma > 0$, Gap(γ)-3-SAT requires $2^{\Omega(n)}$ time

Can we prove Gap-ETH from ETH ?

The above is open, but [A17] provides a sufficient condition for Gap-ETH from cryptographic assumptions

Strong OWF

U_n the uniform distribution over $\{0,1\}^n$

An efficiently computable function $f: \{0,1\}^n \rightarrow \{0,1\}^{m(n)}$ is $(T(n), \varepsilon(n))$ one-way if for all randomized $T(n)$ time algorithm A, $\Pr_{y \sim f(U_n)}[A(y) \in f^{-1}(y)] \leq \varepsilon(n)$ holds

f is an exponentially OWF if f is $(2^{\beta n}, 2^{-\beta n})$ for some $\beta > 0$

f is local if each of its outputs depends on at most k inputs for some constant $k > 0$

Strong OWF

Theorem [A17]

The existence of a family of exponentially-strong locally-computable OWFs implies Gap-ETH

Remark

1. Candidates of such OWFs:

Exponential hardness of random 3-SAT over sparse instances [F02,…], Goldreich's OWF [G11] etc.

2. [A17] shows the existence of a family of exponentially-strong locally-computable PRGs under a similar assumption

Talk Plan

1. Fine-grained complexity
2. Fine-grained complexity and cryptography

[Hardness results]

- ➢ Hardness of SVP/CVP
- ➢ Average-case hardness
- ➢ Gap-ETH from ``strong" OWF

[Easiness results]

- ➢ Beating Brute Force for Systems of Polynomial Equations over Finite Fields

31

References

[LPTWY17] Daniel Lokshtanov, Ramamohan Paturi, Suguru Tamaki, Ryan Williams, Huacheng Yu: Beating Brute Force for Systems of Polynomial Equations over Finite Fields. *SODA* 2017

32

Talk Plan

Part I: **Beating Brute Force for Systems of Polynomial Equations over Finite Fields**

1. Background and Our Results
2. Proof Sketch of Our Results

Part II: **Polynomial Method**

3. Probabilistic Polynomial + majority $\notin$ AC^0[2]
4. Multipoint Polynomial Evaluation

Our Problem: SysPolyEqs(q)

Systems of Multivariate Polynomial Equations over GF[q]

Input:
GF[q] polynomials $p_1, p_2, \dots, p_m$
in formal variables $x_1, x_2, \dots, x_n$

e.g. $q = 3, p_1 = 2x_1^2\, x_2^2\, x_3 + x_3^2\, x_4,\ p_2 = x_1x_2 + x_2^2 + 1$

Task:
find a satisfying assignment $a \in \mathrm{GF}[q]^n$
i.e. $p_1(a) = p_2(a) = \cdots = p_m(a) = 0$ holds

e.g. $(x_1, x_2, x_3, x_4) = (2,2,1,1)$

(#SysPolyEqs(q) denotes the counting version)

Complexity of SysPolyEqs(q)

Input:

GF[q] polynomials $p_1, p_2, \dots, p_m$
in formal variables $x_1, x_2, \dots, x_n$

Parameters: $n, q, k \coloneqq \max \deg(p_i)$

■ P if $k = 1$ (linear equations)

■ NP-complete even if $k = q = 2$

■ k-SAT is a special case of $q = 2$
(each clause can be written as a k-variate polynomial)

■ Best worst-case upper bound: $q^n \times \mathrm{poly}(\text{input-size})$
(even if $q = k = 2$)

SysPolyEqs(q) as Hardness Assumption

Crypto-systems assuming the hardness of:

1. Enumerating all satisfying assignments

■ Hidden Fields Equations (HFE) [Patarin'96,...]

■ Unbalanced Oil and Vinegar signature schemes (UOV) [Kipnis-Patarin-Goubin'99,...]

■ McEliece variants [Faugere-Otmani-Perret-Tillich'10,...]

■ Polly cracker [Albrecht-Faugere-Farshim-Perret'11,...]

…

2. Finding one satisfying assignment

■ QUAD [Berbain-Gilbert-Patarin'06,09,...]

■ Matsumoto-Imai public key scheme [-'88,...]

…

SysPolyEqs(q) as Hardness Assumption

Strong Exponential Time Hypothesis (q^n is necessary) for SysPolyEqs(q) on degree 2 instances implies:

- The current best algorithm for the Listing Triangles problem is optimal [Björklund-Pagh-Vassilevska Williams-Zwick'14]

- Beating brute force for the GF(q)-weight k-clique problem is impossible [Vassilevska-Williams'09]

37

Previous Algorithms

- Groebner Basis: used in practice, double exponential time in the worst case

- $2^{n(1-\epsilon)}$ or polynomial time algorithms for SysPolyEqs(2) on degree 2 instances are known if instances satisfy some conditions e.g. [Yang-Chen'04,Bardet-Faugere-Salvy-Spaenlehauer'13,Miura-Hashimoto-Takagi'13,...]

- $q^{n/2}$ length ``proof" for the unsatisfiability of SysPolyEqs(q) on degree 2 instances [Woods'98] (i.e. nondeterministic algorithm for UNSAT)

38

Our Result 1
[randomized, search, bounded degree]

n variables, GF[q], $k \coloneqq \max \deg(p_i)$

e = 2.718 ... (the base of the natural logarithm)

Condition	Upper Bound
$q = k = 2$	$2^{0.8765n}$
$q = 2, k > 2$	$2^{\left(1-\frac{1}{5k}\right)n}$
$q = p^d, \log p < 4ek$	$q^{\left(1-\frac{1}{200k}\right)n}$
$q = p^d, \log p \geq 4ek$	$q^n \left(\frac{\log q}{edk}\right)^{-dn}$

$\ll q^{\left(1-\frac{1}{O(kq)}\right)n}$

Our Result 2
[deterministic, counting, bounded degree]

n variables, GF[q], $k \coloneqq \max \deg(p_i)$

Condition	Upper Bound
$q = p^d$, k: arbitrary	$q^{\left(1-\frac{1}{300kq^{\frac{6}{7d}}}\right)n}$

Cf. Our Result 1 $q^{\left(1-\frac{1}{200k}\right)n}$

(randomized search)

Generalization of SysPolyEqs(2)

GenSysPolyEqs(2)

Input:

ΣΠΣ circuits (sum of products of linear forms)

$p_1, p_2, \dots, p_m$ in formal variables $x_1, x_2, \dots, x_n$

e.g. $p_1 = \underbrace{(x_1 + x_2 + 1)(x_2 + x_3)}_{} + \underbrace{(x_1 + x_4)x_2}_{} + \underbrace{1}_{}$

Parameters: n, s ≔ total number of products of linear forms

Our Result 3
[GenSysPolyEqs(2), unbounded degree]

n variables, GF[2]

s products of linear forms in total

Type	Upper Bound
Randomized Search	$2^{\left(1-\frac{1}{10\log\left(\frac{s}{n}\right)}\right)n}$
Deterministic Counting	$2^{\left(1-\frac{1}{1100\log\left(\frac{s}{n}\right)}\right)n}$

exponentially faster than 2^n if $s = O(n)$

Remark

(k-)CNF SAT is a special case of SysPolyEqs(2)

(degree k instances)

e.g.

$C_1 = (\neg x_1 \vee x_2 \vee x_3) \Rightarrow p_1 = x_1(1 + x_2)(1 + x_3)$

$C_2 = (x_1 \vee \neg x_3 \vee \neg x_4) \Rightarrow p_2 = (1 + x_1)x_2x_2$

$C_3 = (x_2 \vee x_3 \vee x_4) \Rightarrow p_3 = (1 + x_1)(1 + x_2)(1 + x_3)$

$C_1 = C_2 = C_3 = 1 \Leftrightarrow p_1 = p_2 = p_3 = 0$

Optimality of Our Results

■ SysPolyEqs(2) on degree k instances can be solved in time $2^{n(1-1/O(k))}$

Cf. k-CNF SAT can be solved in time $2^{n(1-1/k)}$

[Paturi-Pudlak-Zane'97,...]

■ For s =the total number of products of linear forms, GenSysPolyEqs(2) can be solved in time $2^{n(1-1/O(\log(s/n)))}$

Cf. For s =the number of clauses,

CNF SAT can be solved in time $2^{n(1-1/(2\log(s/n)))}$

[Schuler'05,Calabro-Impagliazzo-Paturi'06,...]

Proof Sketch for Our Result 1 [randomized, search, bounded degree]

(In what follows, we will focus on GF(2))

Our Techniques

Polynomial Method in Boolean Circuit Complexity plays a key role in recent results:

- Circuit SAT [Williams’11,...]
- All-pairs shortest paths [Williams’14]
- Partial match queries [Abboud-Williams-Yu’15]
- All-points nearest neighbors in Hamming metric [Alman-Williams’15,...]
- Succinct Stable Matching [Moeller-Paturi-Schneider’16]

...

Our Techniques

Two ingredients of our randomized algorithm:

1. fast evaluation algorithms for polynomials [Yates'37,...]

2. approximation of polynomials by low degree probabilistic polynomials [Razborov'87,Smolensky'87] (originally used for proving circuit size lower bounds)

47

Our Tool 1

Lemma 1[Fast Evaluation [Yates'37,...]]

Let $p: \{0,1\}^n \rightarrow \{0,1\}$ be a GF(2)-polynomial represented as a sum of monomials, then, the truth table of p can be generated in time $\mathrm{poly}(n)2^n$

Note:

The number of monomials in p can be 2^n

If we evaluate $p(x)$ for each $x \in \{0,1\}^n$, then it takes $\mathrm{poly}(n)4^n$

48

Basic Idea of Our Algorithm

Input: degree k polynomials $p_1, p_2, \dots, p_m$

1. [Represent as a single polynomial]

   Define $P: \{0,1\}^n \to \{0,1\}$ as $P \coloneqq (1+p_1)\cdots(1+p_m)$

   so that $p_1(x) = p_2(x) = \cdots = p_m(x) = 0 \Leftrightarrow P(x) = 1$

2. [Reduce the number of variables]

   Define $R: \{0,1\}^{n-n'} \to \{0,1\}$ for some $n' < n$ as

   $R(y) \coloneqq \prod_{a \in \{0,1\}^{n'}} (1 + P(y,a))$

   so that $\exists x, P(x) = 1 \Leftrightarrow \exists y, R(y) = 0$

3. [Apply the Fast Evaluation Lemma]

   Get the truth table of $R(y)$ in time $\mathrm{poly}(n) 2^{n-n'}$

Basic Idea Fails

Input: degree k polynomials $p_1, p_2, \dots, p_m$

After Steps 1, 2,

$R(y) = \prod_{a \in \{0,1\}^{n'}} \{1 + \prod_{i=1}^{m} (1 + p_i(y,a))\}$ is such that

$\exists x, p_1(x) = p_2(x) = \cdots = p_m(x) = 0 \Leftrightarrow \exists y, R(y) = 0$

To apply the Fast Evaluation Lemma,
we have to write $R(y)$ as a sum of monomials,
but straightforward expansion needs $2^{n-n'} \times 2^{n'} \approx 2^n$ time

Basic Idea Fails

we have to write $R(y) = \prod_{a \in \{0,1\}^{n'}} \{1 + \prod_{i=1}^{m} (1 + p_i(y,a))\}$

as a sum of monomials, but straightforward expansion needs $2^{n-n'} \times 2^{n'} \approx 2^n$ time

[Expanding inner products]

For each $a \in \{0,1\}^{n'}$, $\prod_{i=1}^{m}(1 + p_i(y,a))$ is a polynomial in $n - n'$ variables $\Longrightarrow$ may have $\approx 2^{n-n'}$ monomials

[Expanding outer products]

We have to multiply such dense polynomials $2^{n'}$ times

Modified Idea: Approximating $R(y)$

by a low degree (i.e. sparse) polynomial

51

Our Tool 2

Definition:

For $S_1, \ldots, S_d \subseteq [m]$,

define a degree d polynomial $Q_{\{s_i\}}: \{0,1\}^m \to \{0,1\}$ as

$Q_{\{s_i\}}(z) \coloneqq \prod_{i=1}^{d}(1 + \sum_{j \in S_i} z_j)$

Intuition: $Q_{\{s_i\}} \approx \prod_{i \in [m]}(1 + z_i) = \neg \bigvee_{i \in [m]} z_i$

Lemma 2[Low-Degree Approximation for NOR

[Razborov'87,...]]

Select random $S_1, \ldots, S_d$ uniformly and independently,

then, for every non-zero $z \in \{0,1\}^m$,

$\Pr[Q_{\{s_i\}}(z) = 0] = 1 - \frac{1}{2^d}$ (cf. $\Pr[Q_{\{s_i\}}(0) = 1] = 1$)

52

Our Tool 2

For random $S_1, \dots, S_d \subseteq [m]$,

$Q_{\{s_i\}}(z) \coloneqq \prod_{i=1}^{d}\left(1 + \sum_{j \in S_i} z_j\right)$

$\Rightarrow \forall z \in \{0,1\}^m, \Pr_{\{s_i\}}[Q_{\{s_i\}}(z) = \prod_{i \in [n]}(1 + z_i)] \geq 1 - \frac{1}{2^d}$

$Q_{\{s_i\}}$ is useful for the following task:
Input: n-variate degree k polynomials $p_1, p_2, \dots, p_m$ $(m \gg n)$
Task: represent $\prod_{i=1}^{m}(1 + p_i)$ as a sum of monomials

The task requires more than 2^n time in general, but
the degree dk polynomial $\prod_{i=1}^{d}\left(1 + \sum_{j \in S_i} p_j\right)$
can be written as a sum of monomials in time $\binom{n}{dk} \ll 2^n$

Modified Basic Idea

Input: degree k polynomials $p_1, p_2, \dots, p_m$

1-2. Define $R: \{0,1\}^{n-n'} \to \{0,1\}$ for some $n' < n$ as

$R(y) \coloneqq \prod_{a \in \{0,1\}^{n'}}\left\{1 + \prod_{i=1}^{m}\left(1 + p_i(y,a)\right)\right\}$ so that

$\exists x, p_1(x) = p_2(x) = \cdots = p_m(x) = 0 \Leftrightarrow \exists y, R(y) = 0$

2.5.a. For each $a \in \{0,1\}^{n'}$, approximate $\prod_{i=1}^{m}(1 + p_i(y,a))$
by $\widehat{P_a} \coloneqq \prod_{i=1}^{d}\left(1 + \sum_{j \in S_i} p_j(y,a)\right)$ with random $S_1, \dots, S_d \subseteq [m]$

2.5.b. Approximate $\prod_{a \in \{0,1\}^{n'}}(1 + \widehat{P_a})$
by $\hat{R} \coloneqq 1 + \sum_{a \in S} \widehat{P_a}$ with random $S \subseteq \{0,1\}^{n'}$

3. [Apply the Fast Evaluation Lemma]
Get the truth table of $\hat{R}(y)$ in time $\mathrm{poly}(n)2^{n-n'}$

Correctness of Our Algorithm

Input: degree k polynomials $p_1, p_2, \dots, p_m$

1-2. Define $R: \{0,1\}^{n-n'} \to \{0,1\}$ for some $n' < n$ as

$R(y) := \prod_{a \in \{0,1\}^{n'}} \{1 + \prod_{i=1}^{m} (1 + p_i(y,a))\}$ so that

$\exists x, p_1(x) = p_2(x) = \cdots = p_m(x) = 0 \Leftrightarrow \exists y, R(y) = 0$

2.5.a. For each $a \in \{0,1\}^{n'}$, approximate $\prod_{i=1}^{m}(1 + p_i(y,a))$

by $\widehat{P_a} := \prod_{i=1}^{d} \left(1 + \sum_{j \in S_i} p_j(y,a)\right)$ with random $S_1, \dots, S_d \subseteq [m]$

2.5.b. Approximate $\prod_{a \in \{0,1\}^{n'}} (1 + \widehat{P_a})$

by $\hat{R} := 1 + \sum_{a \in S} \widehat{P_a}$ with random $S \subseteq \{0,1\}^{n'}$

Correctness: Setting $d - 2 = n'$,

$\forall y \in \{0,1\}^{n-n'}, \Pr[R(y) = \hat{R}(y)] \geq 2/3$

Running Time of Our Algorithm

2.5.a. For each $a \in \{0,1\}^{n'}$, approximate $\prod_{i=1}^{m}(1 + p_i(y,a))$

by $\widehat{P_a} := \prod_{i=1}^{d} \left(1 + \sum_{j \in S_i} p_j(y,a)\right)$ with random $S_1, \dots, S_d \subseteq [m]$

2.5.b. Approximate $\prod_{a \in \{0,1\}^{n'}} (1 + \widehat{P_a})$

by $\hat{R} := 1 + \sum_{a \in S} \widehat{P_a}$ with random $S \subseteq \{0,1\}^{n'}$

3. [Apply the Fast Evaluation Lemma]

Get the truth table of $\hat{R}(y)$ in time $\mathrm{poly}(n) 2^{n-n'}$

[Time for representing a product as a sum of monomials]

each $\widehat{P_a}$ takes $\binom{n-n'}{dk}$ time, $\hat{R}$ takes $\binom{n-n'}{dk} \times 2^{n'}$ time

[Total Running Time] $\binom{n-n'}{dk} \times 2^{n'} + 2^{n-n'} < 2^{\left(1 - \frac{1}{5k}\right)n}$

Proof Sketch for Results 2, 3

Result 2 [deterministic, counting, bounded degree]

Combining Result 1 and

[Derandomization] Epsilon–biased generator [Naor-Naor,...]

[Counting] Modulus amplifying polynomials [Toda, Yao, Beigel-Tarui]

Result 3 [GenSysPolyEqs(2), unbounded degree]

Combining Results 1, 2 and

[Degree reduction] (linear algebraic extension of Schuler's width reduction for CNF):

reduces an instance with s products of linear forms

into a set of SysPolyEqs(2) instances with $k = O(\log(s/n))$

Talk Plan

Part I: **Beating Brute Force for Systems of Polynomial Equations over Finite Fields**

✓ 1. Background and Our Results

✓ 2. Proof Sketch of Our Results

Part II: **Polynomial Method**

3. Probabilistic Polynomial + majority $\notin$ $AC^0[2]$

4. Multipoint Polynomial Evaluation

Probabilistic Polynomial

Definition:

For $S_1, \dots, S_d \subseteq [m]$,

define a degree d polynomial $Q_{\{s_i\}}: \{0,1\}^m \to \{0,1\}$ as

$Q_{\{s_i\}}(z) \coloneqq \prod_{i=1}^{d} \left(1 + \sum_{j \in S_i} z_j\right)$

Intuition: $Q_{\{s_i\}} \approx \prod_{i \in [m]} (1 + z_i) = \neg \bigvee_{i \in [m]} z_i$

Lemma 2[Low-Degree Approximation for NOR

[Razborov'87,...]]

Select random $S_1, \dots, S_d$ uniformly and independently,

then, for every non-zero $z \in \{0,1\}^m$,

$\Pr[Q_{\{s_i\}}(z) = 0] = 1 - \frac{1}{2^d}$ (cf. $\Pr[Q_{\{s_i\}}(0) = 1] = 1$)

59

Proof of Lemma 2

For random $S_1, \dots, S_d \subseteq [m]$, $Q_{\{s_i\}}(z) \coloneqq \prod_{i=1}^{d} \left(1 + \sum_{j \in S_i} z_j\right)$

Then, for every non-zero $z \in \{0,1\}^m$,

$\Pr[Q_{\{s_i\}}(z) = 0] = 1 - \frac{1}{2^d}$ (cf. $\Pr[Q_{\{s_i\}}(0) = 1] = 1$)

$\mathrm{z} = 0 \Rightarrow \forall i, \Pr[1 + \sum_{j \in S_i} z_j = 1] = 1 \Rightarrow \Pr[Q_{\{s_i\}}(0) = 1] = 1$

60

Proof of Lemma 2

For random $S_1, \dots, S_d \subseteq [m]$, $Q_{\{s_i\}}(z) \coloneqq \prod_{i=1}^{d}\left(1 + \sum_{j \in S_i} z_j\right)$

Then, for every non-zero $z \in \{0,1\}^m$,

$\Pr[Q_{\{s_i\}}(z) = 0] = 1 - \frac{1}{2^d}$ (cf. $\Pr[Q_{\{s_i\}}(0) = 1] = 1$)

1. $z \neq 0 \Rightarrow \forall i, \Pr[1 + \sum_{j \in S_i} z_j = 0] = \Pr[1 + \sum_{j \in S_i} z_j = 1] = \frac{1}{2}$
2. $Q_{\{s_i\}}(z) = 1 \ \Leftrightarrow \forall i, 1 + \sum_{j \in S_i} z_j = 1$

1+2 $\Rightarrow \Pr\left[Q_{\{s_i\}}(z) = 0\right] = 1 - \Pr\left[Q_{\{s_i\}}(z) = 1\right]$

$= 1 - \Pr\left[\forall i, 1 + \sum_{j \in S_i} z_j = 1\ \right] = 1 - \frac{1}{2^d}$

majority ∉ AC⁰[2]

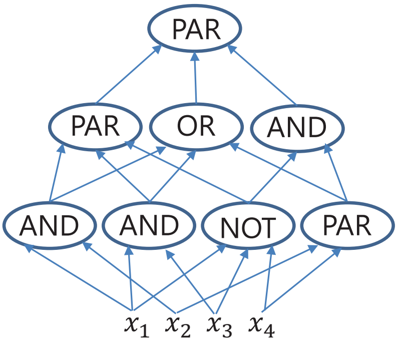
AC^0[2] circuit:

gate set = {AND, OR, NOT, PARITY}

(unbounded fan-in/out)

depth $= k = O(1)$

size (#gates) $= s$



Theorem [Razborov'87]

An AC^0[2] circuit computes the n-variate majority function

$\Rightarrow s = 2^{\Omega\left(n^{\frac{1}{2k}}\right)}$

Proof of majority $\notin$ AC^0[2]

Lemma 3

C: AC^0[2] circuit, depth $= k = O(1)$, size (#gates) $= s$

$\Rightarrow$ $\exists$random polynomial P of degree $d = O(\log^k s)$ s.t.

$\forall x \in \{0,1\}^n, \Pr_P[C(x) = P(x)] \geq 0.999$

Proof Sketch: Replace each

- NOT(y) by $1 + y$
- PARITY$(y_1, y_2, \ldots)$ by $y_1 + y_2 + \cdots$
- AND/OR by Low-Degree Approximation for NOR of Lemma 2 with De Morgan's Law

$P(x)$ is a composition of polynomials = a single polynomial

the union bound $\Rightarrow$ error probability

Proof of majority $\notin$ AC^0[2]

Lemma 3

C: AC^0[2] circuit, depth $= k = O(1)$, size (#gates) $= s$

$\Rightarrow$ $\exists$random polynomial P of degree $d = O(\log^k s)$ s.t.

$\forall x \in \{0,1\}^n, \Pr_P[C(x) = P(x)] \geq 0.999$

$\Rightarrow$ $\exists$polynomial p of degree $d = O(\log^k s)$ s.t.

$\Pr_{x \in \{0,1\}^n}[C(x) = p(x)] \geq 0.999$

Proof of majority $\notin$ AC^0[2]

Corollary of Lemma 3

C: AC^0[2] circuit, depth $= k = O(1)$, size (#gates) $= s$

$\Rightarrow$ $\exists$polynomial p of degree $d = O(\log^k s)$ s.t.

$\Pr_{x\in\{0,1\}^n}[C(x) = p(x)] \geq 0.999$

Lemma 4

$\forall$polynomial p of degree $d = o(\sqrt{n})$

$\Pr_{x\in\{0,1\}^n}[p(x) = \text{majority}(x)] < 2/3$

$s = 2^{o(n^{\frac{1}{2k}})}$ $\Rightarrow$ $\exists$polynomial p of degree $d = o(\sqrt{n})$ s.t.

$\Pr_{x\in\{0,1\}^n}[C(x) = p(x)] \geq 0.999$

$\Rightarrow \Pr_{x\in\{0,1\}^n}[C(x) = \text{majority}(x)] < 2/3 + 0.001$

Talk Plan

Part I: **Beating Brute Force for Systems of Polynomial Equations over Finite Fields**

✓ 1. Background and Our Results

✓ 2. Proof Sketch of Our Results

Part II: **Polynomial Method**

✓ 3. Probabilistic Polynomial + majority $\notin$ AC^0[2]

4. Multipoint Polynomial Evaluation

Multipoint Polynomial Evaluation

Lemma 1[Fast Evaluation [Yates'37,...]]

Let $p:\{0,1\}^n \to \{0,1\}$ be a GF(2)-polynomial represented as a sum of monomials, then, the truth table of p can be generated in time $\mathrm{poly}(n)2^n$

Note:

The number of monomials in p can be 2^n

If we evaluate $p(x)$ for each $x \in \{0,1\}^n$, then it takes $\mathrm{poly}(n)4^n$

Proof of Lemma 1

Lemma 1[Fast Evaluation [Yates'37,...]]

Let $p:\{0,1\}^n \to \{0,1\}$ be a GF(2)-polynomial represented as a sum of monomials, then, the truth table of p can be generated in time $\mathrm{poly}(n)2^n$

Several Proofs are known:

1. Dynamic Programming
2. Fast Fourier Transform
3. Fast Rectangular Matrix Multiplication

Proof of Lemma 1

Lemma 1′[Fast Evaluation]

Let $p\colon \{0,1\}^n \rightarrow \{0,1\}$ be a GF(2)-polynomial represented as a sum of $O(2^{n/7})$ monomials, then, the truth table of p can be generated in time $\mathrm{poly}(n)2^n$

3. Fast Rectangular Matrix Multiplication

A: $N \times N^{0.3}$ matrix, B: $N^{0.3} \times N$ matrix

$C = AB$ can be obtained in $O(N^2)$ time

[Coppersmith, Le Gall, …]

Proof of Lemma 1

Example: $p(x_{1,}x_2, y_1, y_2) = x_1 + y_1y_2 + x_1x_2y_1 + x_1x_2y_1y_2$

Want: $2^2 \times 2^2$ matrix $p(00{,}00)$, $p(00{,}01)$,…, $p(11{,}11)$

Observation:

$A\left(x_{1,}x_2\right) \coloneqq (x_1, 1, x_1x_2, x_1x_2), B(y_1, y_2) \coloneqq (1, y_1y_2, y_1, y_1y_2)^t$

$\Rightarrow p(x_{1,}x_2, y_1, y_2) = A\left(x_{1,}x_2\right)B(y_1, y_2)$

$$A \coloneqq \begin{pmatrix} A(00) \\ A(01) \\ A(10) \\ A(11) \end{pmatrix}, B \coloneqq (B(00), B(01), B(10), B(11))$$

$$\Rightarrow AB = \begin{pmatrix} p(00{,}00) & \cdots & p(00{,}11) \\ \vdots & \ddots & \vdots \\ p(11{,}00) & \cdots & p(11{,}11) \end{pmatrix}$$

Proof of Lemma 1

Let $p\colon \{0,1\}^n \to \{0,1\}$ be a GF(2)-polynomial represented as a sum of $O(2^{n/7})$ monomials

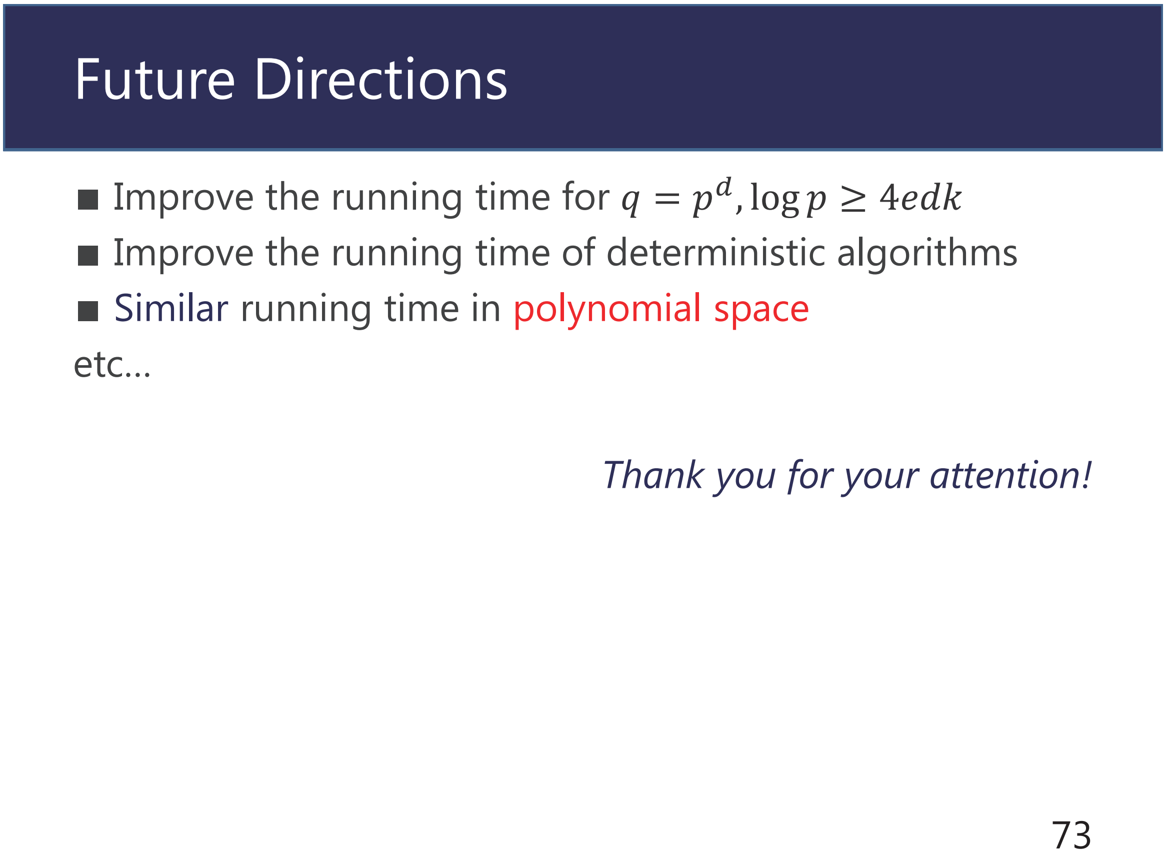
Let $x_1, \ldots, x_{n/2}, y_1, \ldots, y_{n/2}$ be formal variables
Then, for $r = O(2^{n/7})$,
$p\left(x_1, \ldots, x_{n/2}, y_1, \ldots, y_{n/2}\right) = \sum_{i=1}^{r} f_i(x) g_i(y)$

Construct $2^{n/2} \times r$ and $r \times 2^{n/2}$ matrices
$A_{x,i} \coloneqq f_i(x), B_{i,y} \coloneqq g_i(y)$
$\Rightarrow (AB)_{x,y} = \sum_{i=1}^{r} f_i(x) g_i(y)$
(multiplication of AB in $O(2^n)$ time)

71

Conclusion

72



Future Directions

- Improve the running time for $q = p^d, \log p \geq 4edk$
- Improve the running time of deterministic algorithms
- Similar running time in polynomial space

etc…

Thank you for your attention!

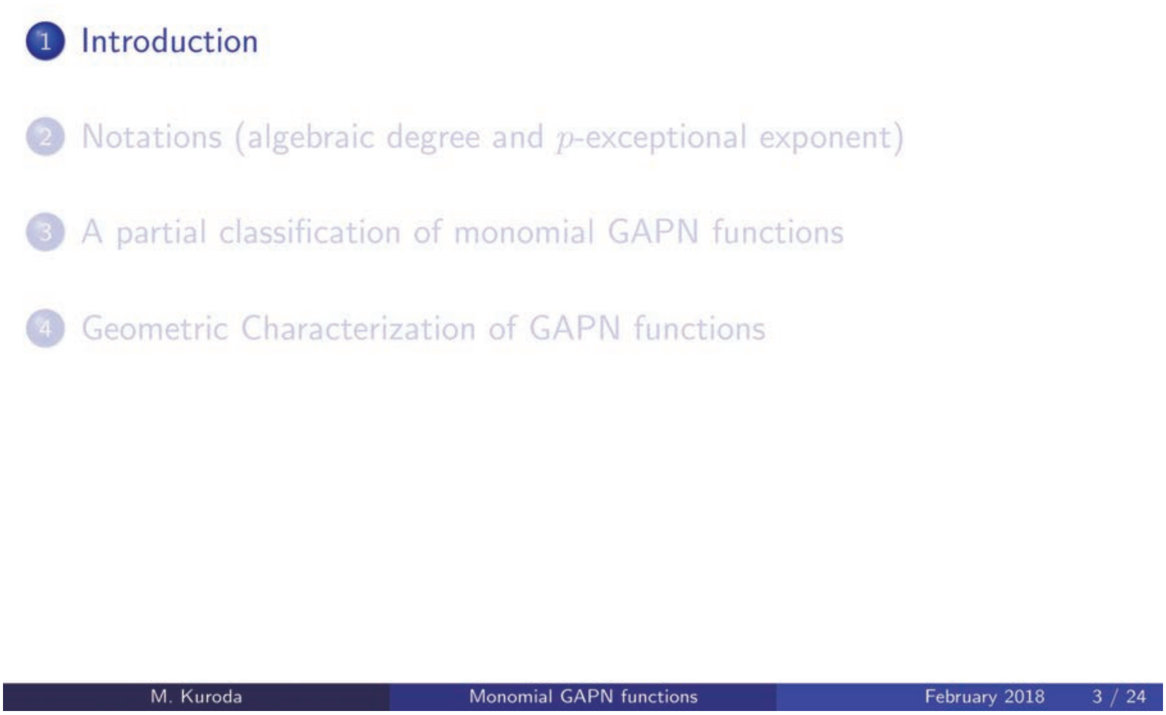
On monomial GAPN (Generalized Almost Perfect Nonlinear) functions and their classification

Masamichi Kuroda
(joint work with Shuhei Tsujie)

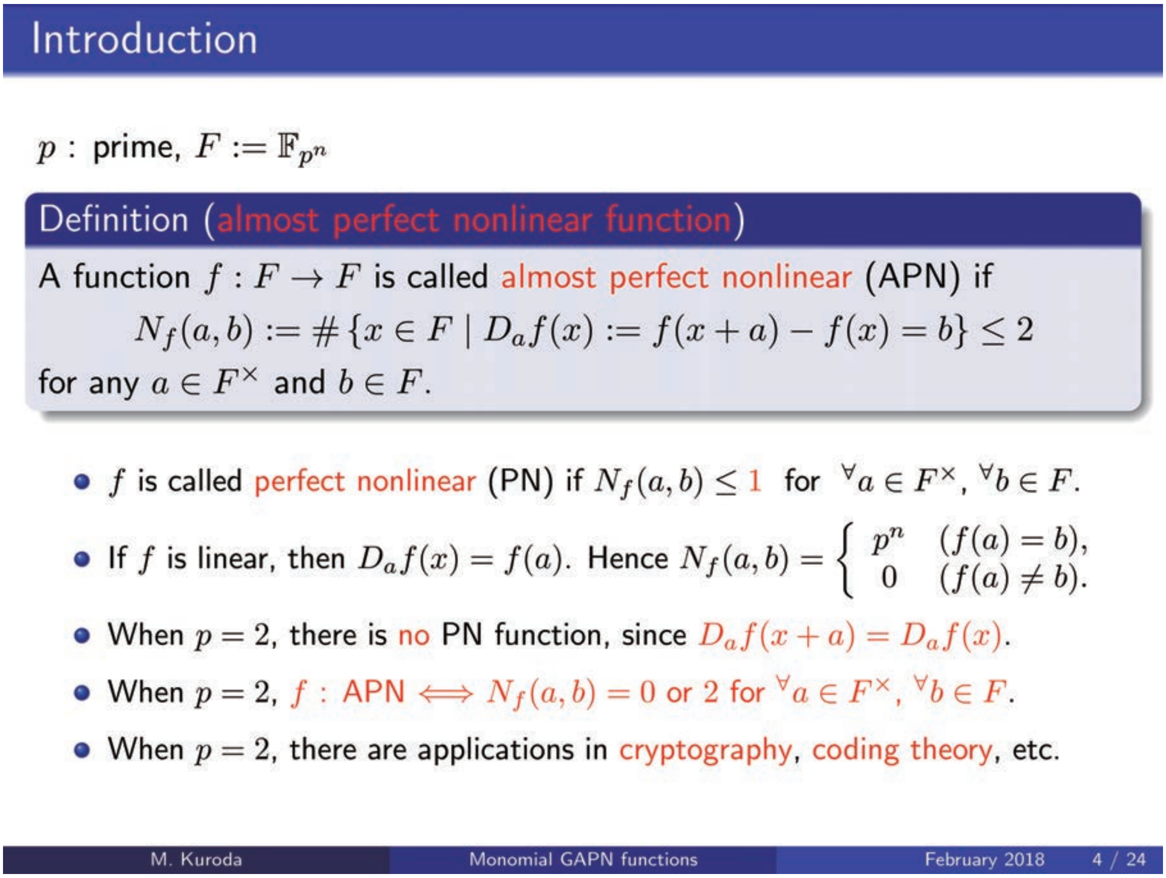
at Institute of Mathematics for Industry
Kyushu University
February 6, 2018

Contents

1. Introduction
2. Notations (algebraic degree and p-exceptional exponent)
3. A partial classification of monomial GAPN functions
4. Geometric Characterization of GAPN functions



1. Introduction
2. Notations (algebraic degree and p-exceptional exponent)
3. A partial classification of monomial GAPN functions
4. Geometric Characterization of GAPN functions



Introduction

p : prime, $F := \mathbb{F}_{p^n}$

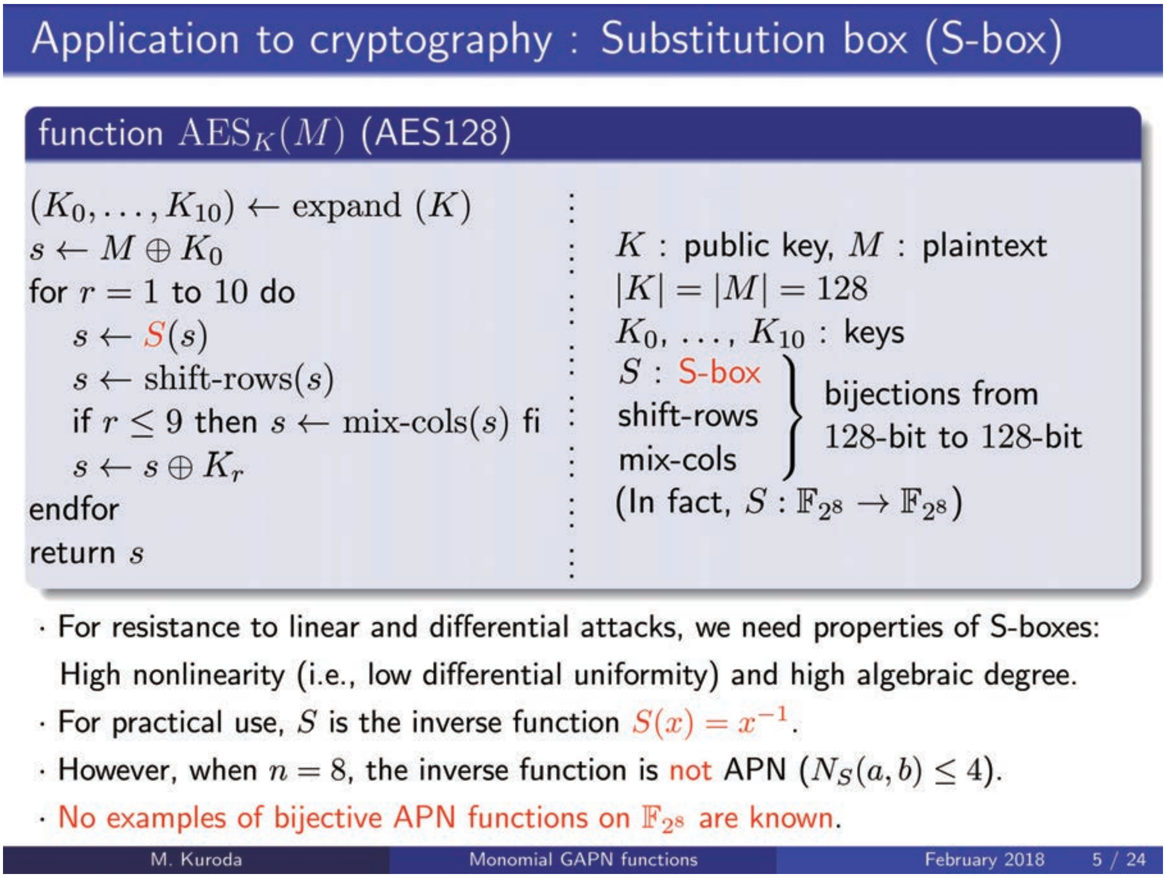
Definition (almost perfect nonlinear function)

A function $f : F \to F$ is called almost perfect nonlinear (APN) if

$$N_f(a,b) := \#\{x \in F \mid D_a f(x) := f(x+a) - f(x) = b\} \leq 2$$

for any $a \in F^\times$ and $b \in F$.

- f is called perfect nonlinear (PN) if $N_f(a,b) \leq 1$ for ${}^\forall a \in F^\times$, ${}^\forall b \in F$.
- If f is linear, then $D_a f(x) = f(a)$. Hence $N_f(a,b) = \begin{cases} p^n & (f(a) = b), \\ 0 & (f(a) \neq b). \end{cases}$
- When $p = 2$, there is no PN function, since $D_a f(x+a) = D_a f(x)$.
- When $p = 2$, f : APN $\Longleftrightarrow$ $N_f(a,b) = 0$ or 2 for ${}^\forall a \in F^\times$, ${}^\forall b \in F$.
- When $p = 2$, there are applications in cryptography, coding theory, etc.



Application to cryptography : Substitution box (S-box)

function $\mathrm{AES}_K(M)$ (AES128)

```
$(K_0, \dots, K_{10}) \leftarrow \mathrm{expand}\ (K)$
$s \leftarrow M \oplus K_0$
for $r = 1$ to $10$ do
    $s \leftarrow S(s)$
    $s \leftarrow \mathrm{shift\text{-}rows}(s)$
    if $r \leq 9$ then $s \leftarrow \mathrm{mix\text{-}cols}(s)$ fi
    $s \leftarrow s \oplus K_r$
endfor
return $s$
```

K : public key, M : plaintext

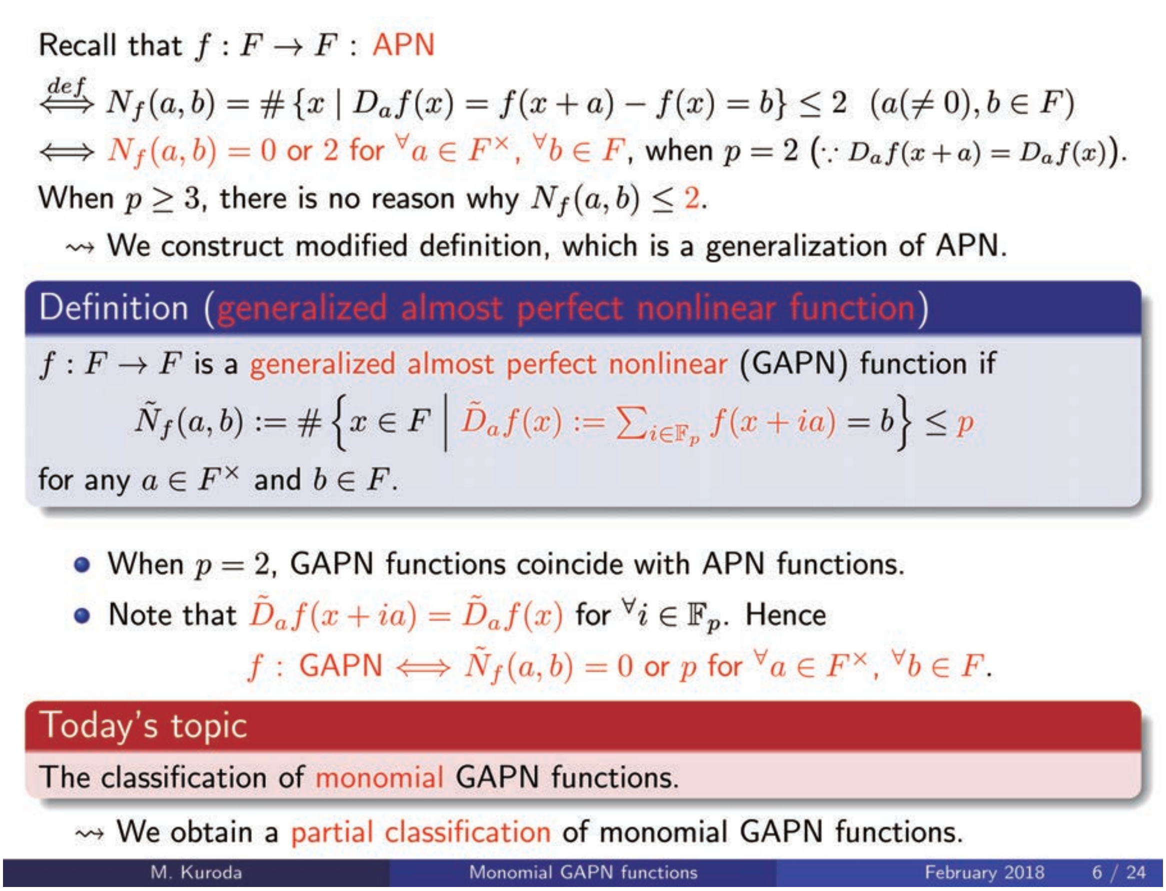
$|K| = |M| = 128$

$K_0, \dots, K_{10}$: keys

$\left.\begin{array}{l} S : \text{S-box} \\ \text{shift-rows} \\ \text{mix-cols} \end{array}\right\}$ bijections from 128-bit to 128-bit

(In fact, $S : \mathbb{F}_{2^8} \to \mathbb{F}_{2^8}$)

- For resistance to linear and differential attacks, we need properties of S-boxes: High nonlinearity (i.e., low differential uniformity) and high algebraic degree.
- For practical use, S is the inverse function $S(x) = x^{-1}$.
- However, when $n = 8$, the inverse function is not APN ($N_S(a, b) \leq 4$).
- No examples of bijective APN functions on $\mathbb{F}_{2^8}$ are known.



Recall that $f : F \to F$: APN

$\overset{def}{\Longleftrightarrow} N_f(a, b) = \#\{x \mid D_a f(x) = f(x + a) - f(x) = b\} \leq 2 \ \ (a(\neq 0), b \in F)$

$\Longleftrightarrow N_f(a, b) = 0$ or 2 for ${}^\forall a \in F^\times$, ${}^\forall b \in F$, when $p = 2$ ($\because D_a f(x + a) = D_a f(x)$).

When $p \geq 3$, there is no reason why $N_f(a, b) \leq 2$.

⇝ We construct modified definition, which is a generalization of APN.

Definition (generalized almost perfect nonlinear function)

$f : F \to F$ is a generalized almost perfect nonlinear (GAPN) function if

$$\tilde{N}_f(a, b) := \#\left\{x \in F \,\middle|\, \tilde{D}_a f(x) := \textstyle\sum_{i \in \mathbb{F}_p} f(x + ia) = b\right\} \leq p$$

for any $a \in F^\times$ and $b \in F$.

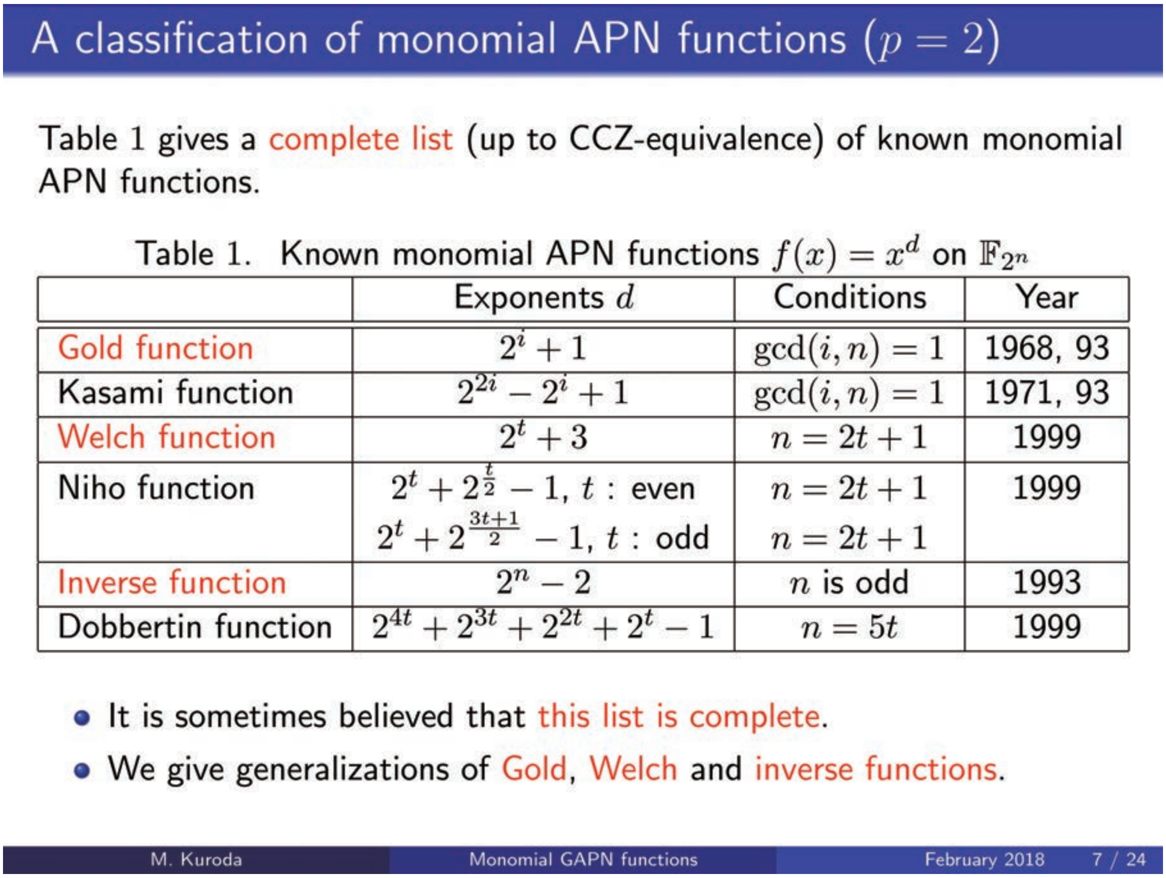
- When $p = 2$, GAPN functions coincide with APN functions.
- Note that $\tilde{D}_a f(x + ia) = \tilde{D}_a f(x)$ for ${}^\forall i \in \mathbb{F}_p$. Hence

$$f : \text{GAPN} \Longleftrightarrow \tilde{N}_f(a, b) = 0 \text{ or } p \text{ for } {}^\forall a \in F^\times,\ {}^\forall b \in F.$$

Today's topic

The classification of monomial GAPN functions.

⇝ We obtain a partial classification of monomial GAPN functions.



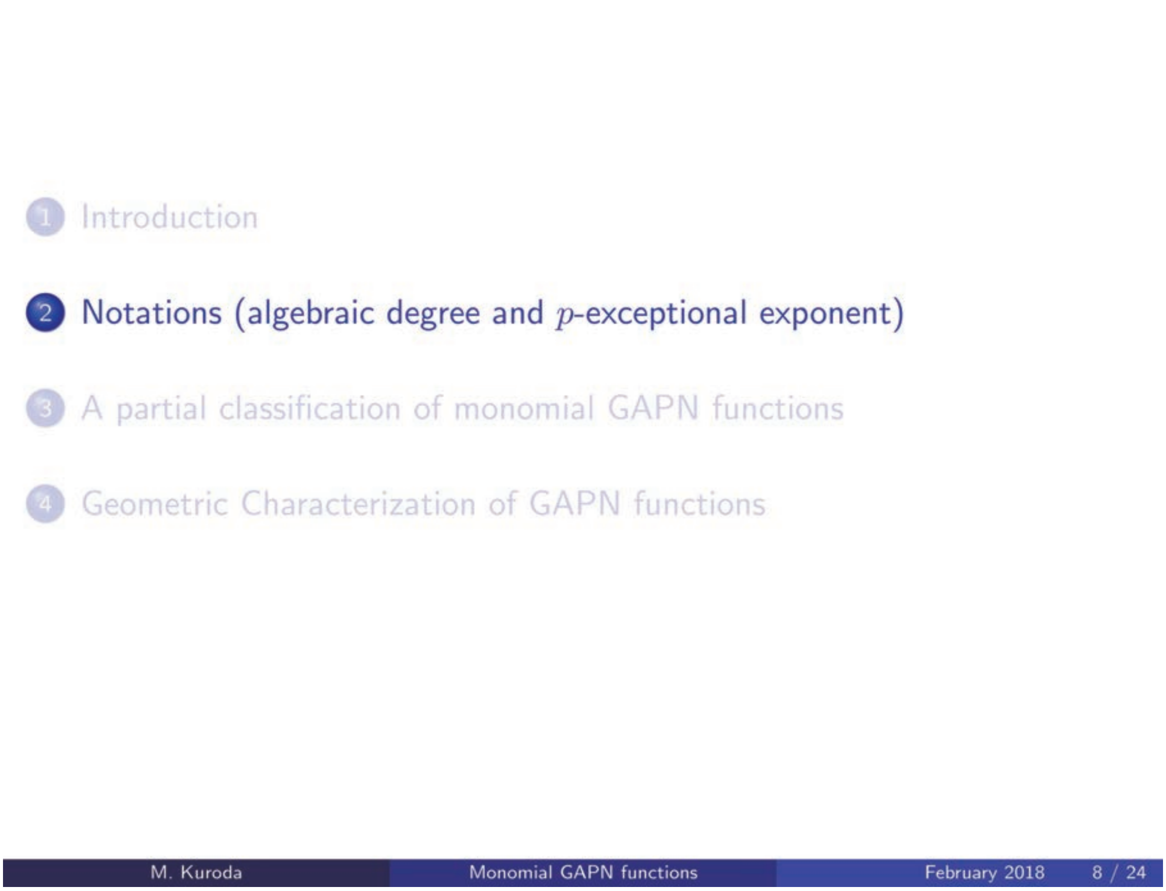
A classification of monomial APN functions ($p = 2$)

Table 1 gives a complete list (up to CCZ-equivalence) of known monomial APN functions.

Table 1. Known monomial APN functions $f(x) = x^d$ on $\mathbb{F}_{2^n}$

	Exponents d	Conditions	Year
Gold function	$2^i + 1$	$\gcd(i, n) = 1$	1968, 93
Kasami function	$2^{2i} - 2^i + 1$	$\gcd(i, n) = 1$	1971, 93
Welch function	$2^t + 3$	$n = 2t + 1$	1999
Niho function	$2^t + 2^{\frac{t}{2}} - 1$, t : even $2^t + 2^{\frac{3t+1}{2}} - 1$, t : odd	$n = 2t + 1$ $n = 2t + 1$	1999
Inverse function	$2^n - 2$	n is odd	1993
Dobbertin function	$2^{4t} + 2^{3t} + 2^{2t} + 2^t - 1$	$n = 5t$	1999

- It is sometimes believed that this list is complete.
- We give generalizations of Gold, Welch and inverse functions.

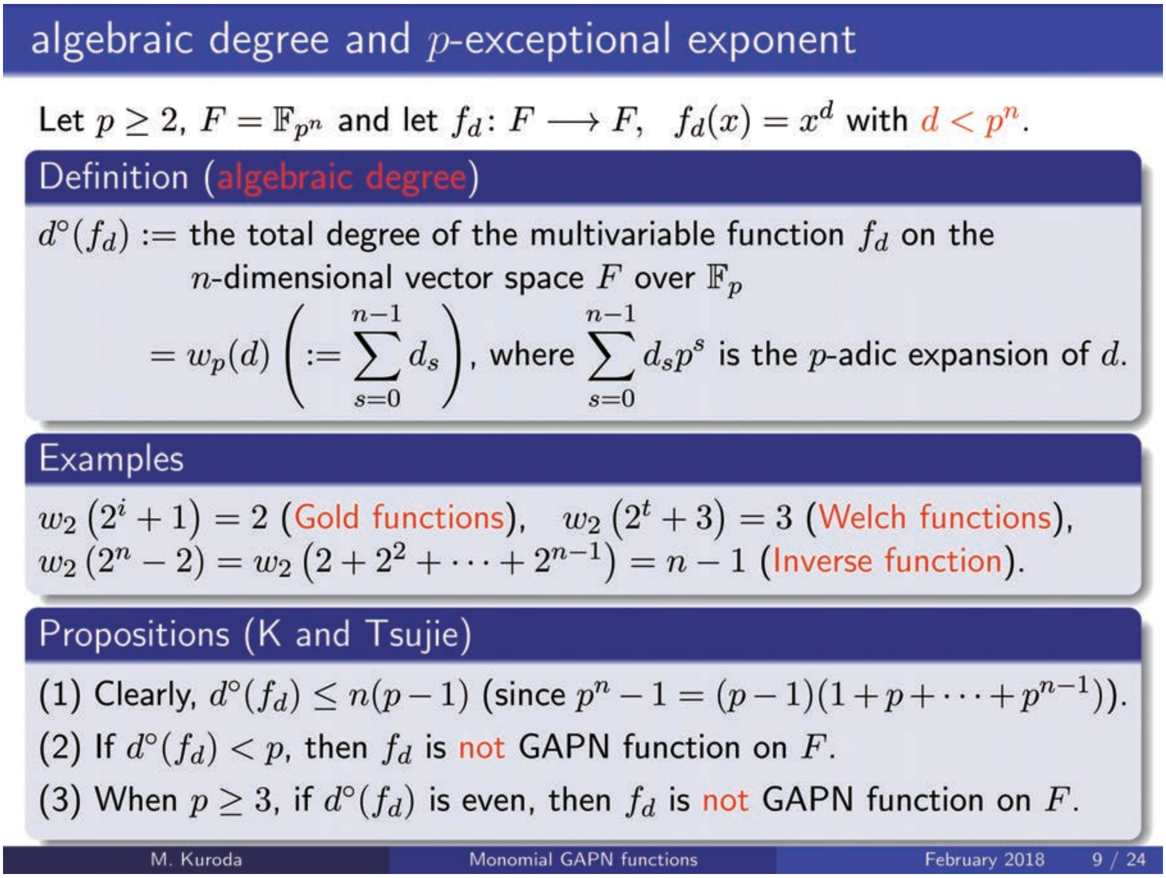


1 Introduction

2 Notations (algebraic degree and p-exceptional exponent)

3 A partial classification of monomial GAPN functions

4 Geometric Characterization of GAPN functions



algebraic degree and p-exceptional exponent

Let $p \geq 2$, $F = \mathbb{F}_{p^n}$ and let $f_d \colon F \longrightarrow F$, $f_d(x) = x^d$ with $d < p^n$.

Definition (algebraic degree)

$$d^\circ(f_d) := \text{the total degree of the multivariable function } f_d \text{ on the } n\text{-dimensional vector space } F \text{ over } \mathbb{F}_p$$
$$= w_p(d) \left(:= \sum_{s=0}^{n-1} d_s \right), \text{ where } \sum_{s=0}^{n-1} d_s p^s \text{ is the } p\text{-adic expansion of } d.$$

Examples

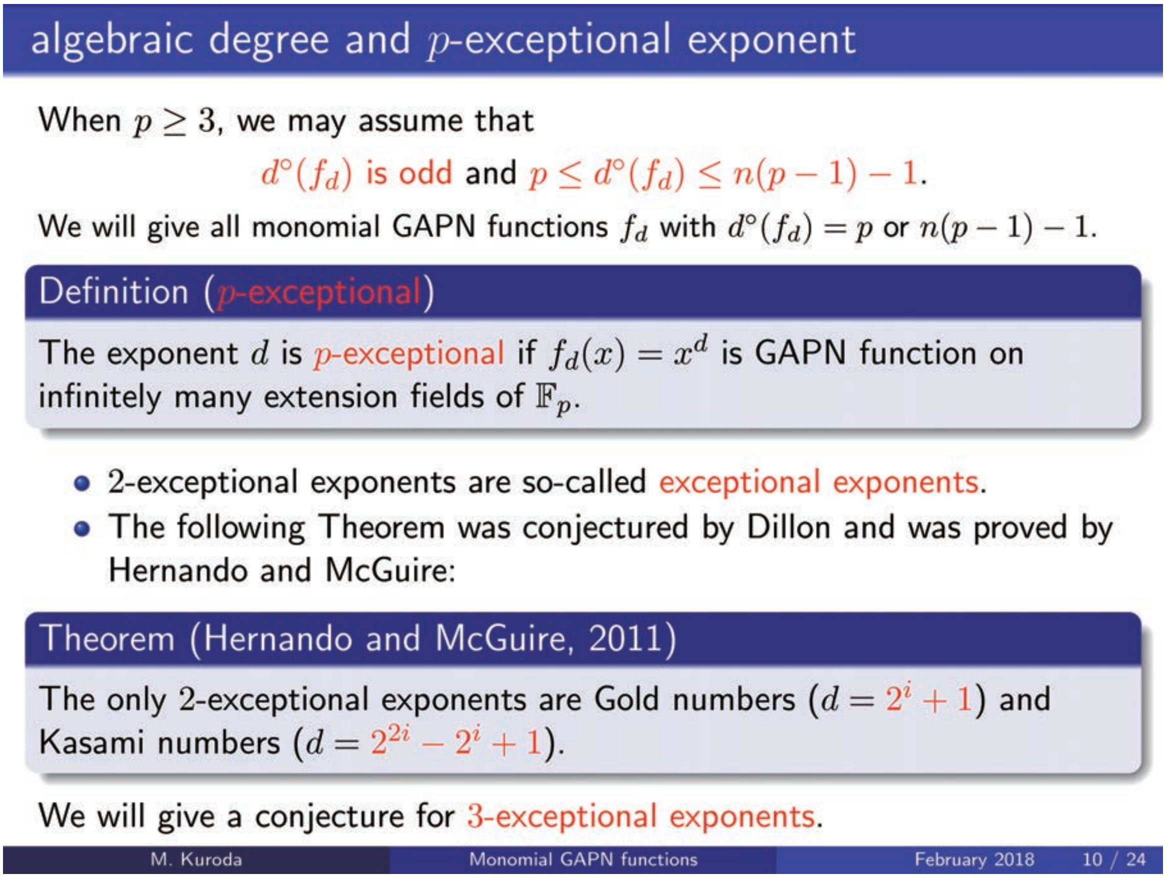
$w_2\left(2^i + 1\right) = 2$ (Gold functions), $w_2\left(2^t + 3\right) = 3$ (Welch functions),
$w_2\left(2^n - 2\right) = w_2\left(2 + 2^2 + \cdots + 2^{n-1}\right) = n - 1$ (Inverse function).

Propositions (K and Tsujie)

(1) Clearly, $d^\circ(f_d) \leq n(p-1)$ (since $p^n - 1 = (p-1)(1 + p + \cdots + p^{n-1})$).

(2) If $d^\circ(f_d) < p$, then f_d is not GAPN function on F.

(3) When $p \geq 3$, if $d^\circ(f_d)$ is even, then f_d is not GAPN function on F.



algebraic degree and p-exceptional exponent

When $p \geq 3$, we may assume that

$$d^\circ(f_d) \text{ is odd and } p \leq d^\circ(f_d) \leq n(p-1) - 1.$$

We will give all monomial GAPN functions f_d with $d^\circ(f_d) = p$ or $n(p-1) - 1$.

Definition (p-exceptional)

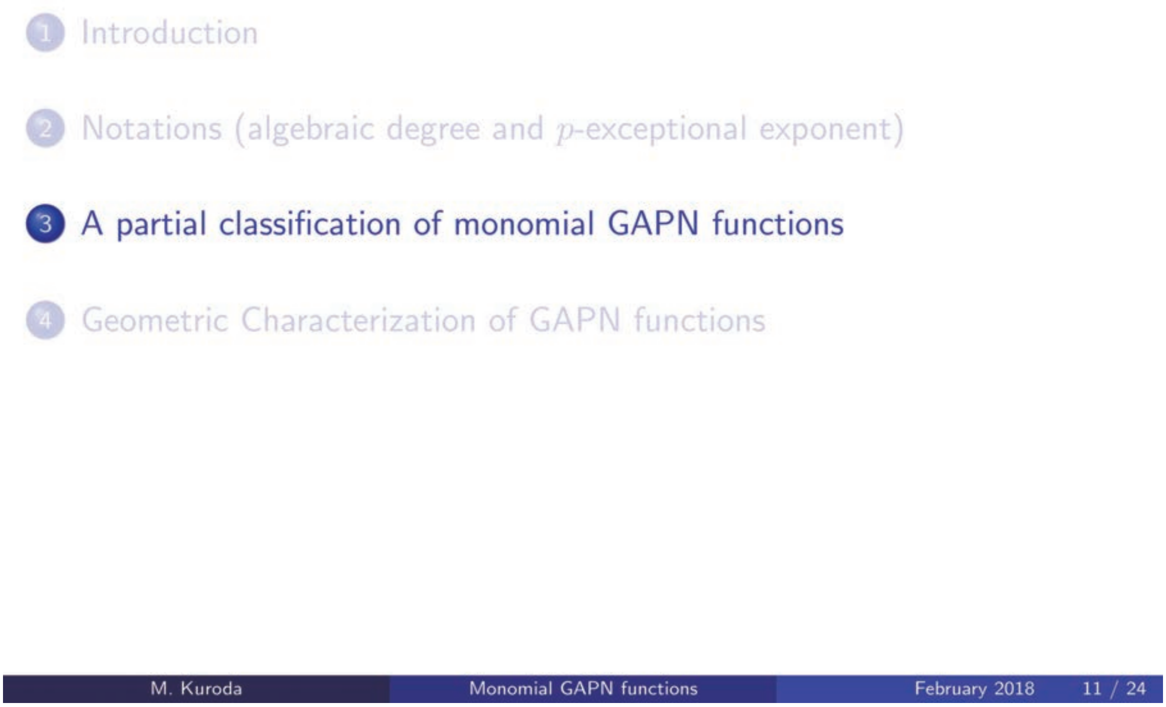
The exponent d is p-exceptional if $f_d(x) = x^d$ is GAPN function on infinitely many extension fields of $\mathbb{F}_p$.

- 2-exceptional exponents are so-called exceptional exponents.
- The following Theorem was conjectured by Dillon and was proved by Hernando and McGuire:

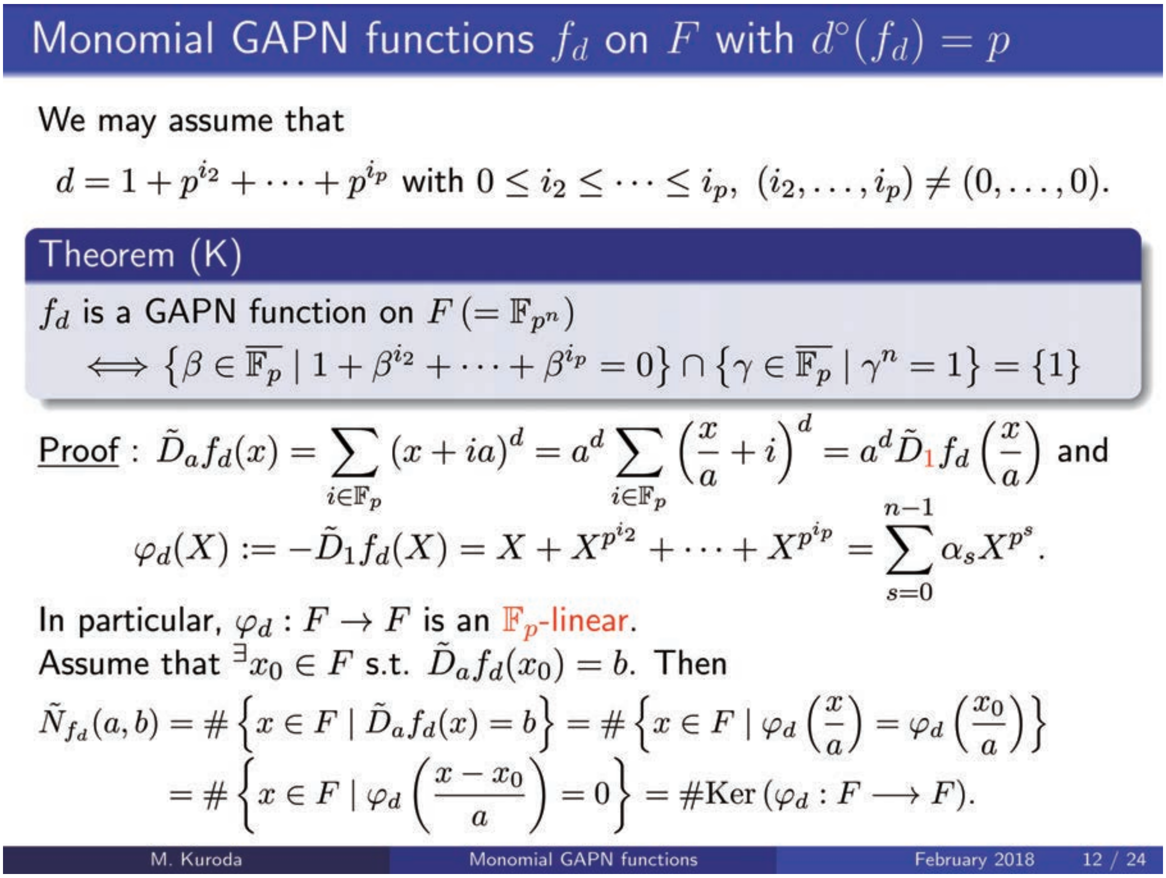
Theorem (Hernando and McGuire, 2011)

The only 2-exceptional exponents are Gold numbers ($d = 2^i + 1$) and Kasami numbers ($d = 2^{2i} - 2^i + 1$).

We will give a conjecture for 3-exceptional exponents.



1. Introduction
2. Notations (algebraic degree and p-exceptional exponent)
3. A partial classification of monomial GAPN functions
4. Geometric Characterization of GAPN functions



Monomial GAPN functions f_d on F with $d^\circ(f_d) = p$

We may assume that

$$d = 1 + p^{i_2} + \cdots + p^{i_p} \text{ with } 0 \le i_2 \le \cdots \le i_p,\ (i_2, \ldots, i_p) \ne (0, \ldots, 0).$$

Theorem (K)

f_d is a GAPN function on $F\,(= \mathbb{F}_{p^n})$

$$\Longleftrightarrow \left\{\beta \in \overline{\mathbb{F}_p} \mid 1 + \beta^{i_2} + \cdots + \beta^{i_p} = 0\right\} \cap \left\{\gamma \in \overline{\mathbb{F}_p} \mid \gamma^n = 1\right\} = \{1\}$$

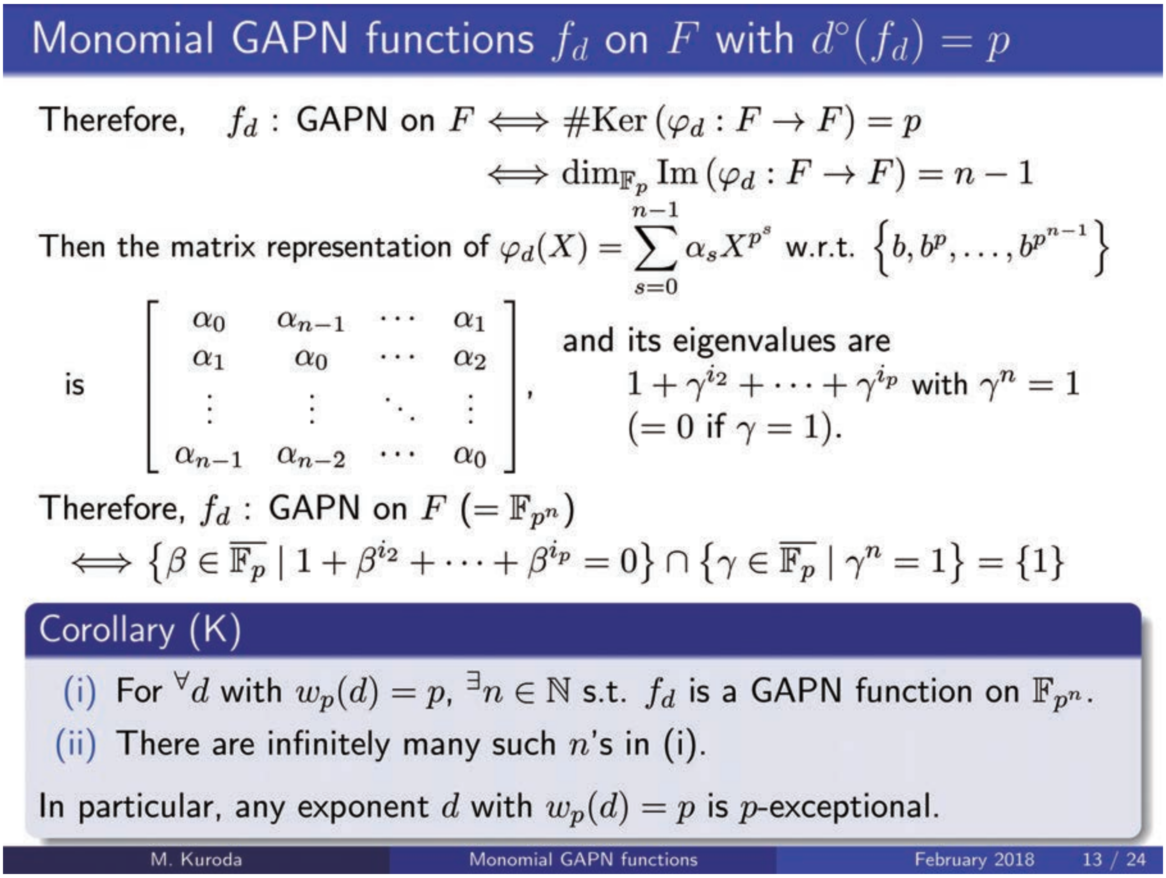
Proof : $\tilde{D}_a f_d(x) = \sum_{i \in \mathbb{F}_p} (x + ia)^d = a^d \sum_{i \in \mathbb{F}_p} \left(\frac{x}{a} + i\right)^d = a^d \tilde{D}_1 f_d\left(\frac{x}{a}\right)$ and

$$\varphi_d(X) := -\tilde{D}_1 f_d(X) = X + X^{p^{i_2}} + \cdots + X^{p^{i_p}} = \sum_{s=0}^{n-1} \alpha_s X^{p^s}.$$

In particular, $\varphi_d : F \to F$ is an $\mathbb{F}_p$-linear.
Assume that $^\exists x_0 \in F$ s.t. $\tilde{D}_a f_d(x_0) = b$. Then

$$\tilde{N}_{f_d}(a, b) = \#\left\{x \in F \mid \tilde{D}_a f_d(x) = b\right\} = \#\left\{x \in F \mid \varphi_d\left(\frac{x}{a}\right) = \varphi_d\left(\frac{x_0}{a}\right)\right\}$$

$$= \#\left\{x \in F \mid \varphi_d\left(\frac{x - x_0}{a}\right) = 0\right\} = \#\mathrm{Ker}\,(\varphi_d : F \longrightarrow F).$$



Monomial GAPN functions f_d on F with $d^\circ(f_d) = p$

Therefore, f_d : GAPN on $F \iff \#\mathrm{Ker}\,(\varphi_d : F \to F) = p$

$$\iff \dim_{\mathbb{F}_p} \mathrm{Im}\,(\varphi_d : F \to F) = n-1$$

Then the matrix representation of $\varphi_d(X) = \sum_{s=0}^{n-1} \alpha_s X^{p^s}$ w.r.t. $\left\{b, b^p, \dots, b^{p^{n-1}}\right\}$

is $\begin{bmatrix} \alpha_0 & \alpha_{n-1} & \cdots & \alpha_1 \\ \alpha_1 & \alpha_0 & \cdots & \alpha_2 \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{n-1} & \alpha_{n-2} & \cdots & \alpha_0 \end{bmatrix}$, and its eigenvalues are $1 + \gamma^{i_2} + \cdots + \gamma^{i_p}$ with $\gamma^n = 1$ $(= 0$ if $\gamma = 1)$.

Therefore, f_d : GAPN on F $(= \mathbb{F}_{p^n})$

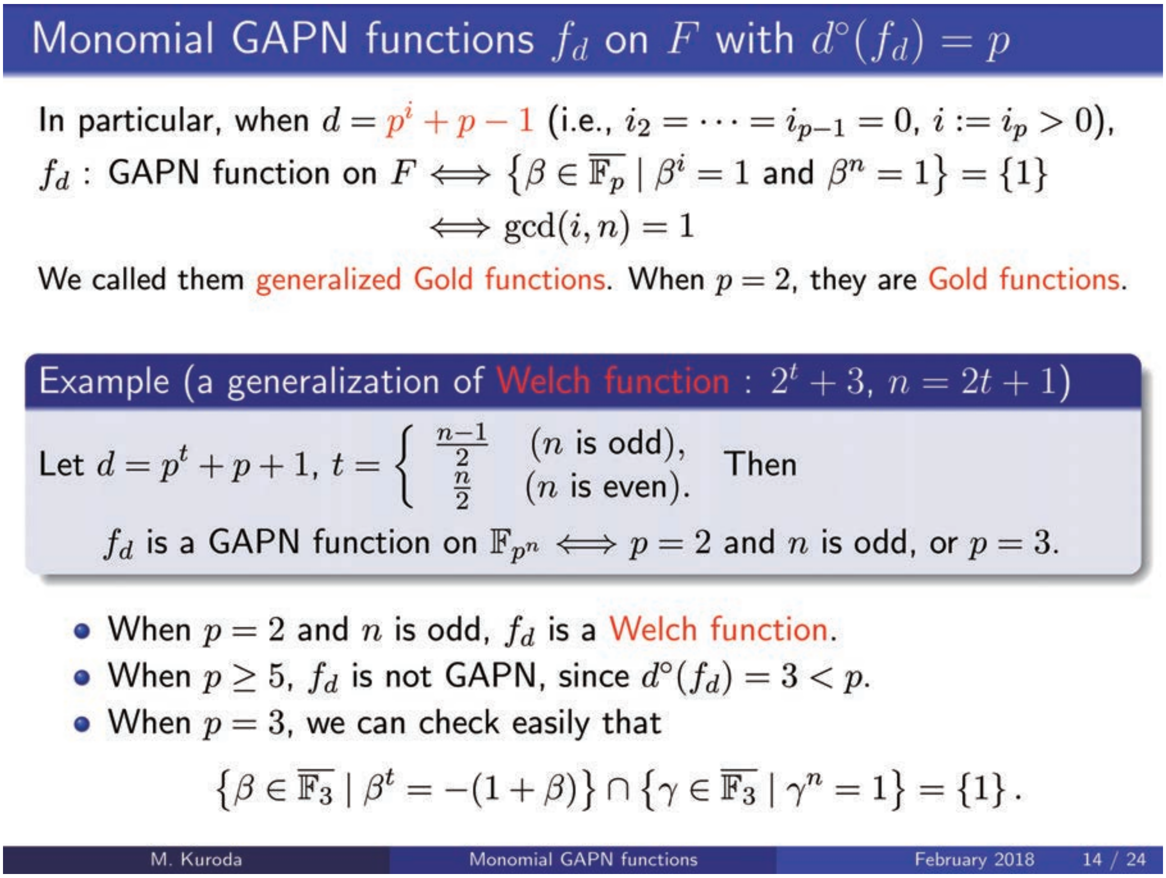
$$\iff \{\beta \in \overline{\mathbb{F}_p} \mid 1 + \beta^{i_2} + \cdots + \beta^{i_p} = 0\} \cap \{\gamma \in \overline{\mathbb{F}_p} \mid \gamma^n = 1\} = \{1\}$$

Corollary (K)

(i) For $^\forall d$ with $w_p(d) = p$, $^\exists n \in \mathbb{N}$ s.t. f_d is a GAPN function on $\mathbb{F}_{p^n}$.

(ii) There are infinitely many such n's in (i).

In particular, any exponent d with $w_p(d) = p$ is p-exceptional.



Monomial GAPN functions f_d on F with $d^\circ(f_d) = p$

In particular, when $d = p^i + p - 1$ (i.e., $i_2 = \cdots = i_{p-1} = 0$, $i := i_p > 0$),

f_d : GAPN function on $F \iff \{\beta \in \overline{\mathbb{F}_p} \mid \beta^i = 1$ and $\beta^n = 1\} = \{1\}$

$$\iff \gcd(i, n) = 1$$

We called them generalized Gold functions. When $p = 2$, they are Gold functions.

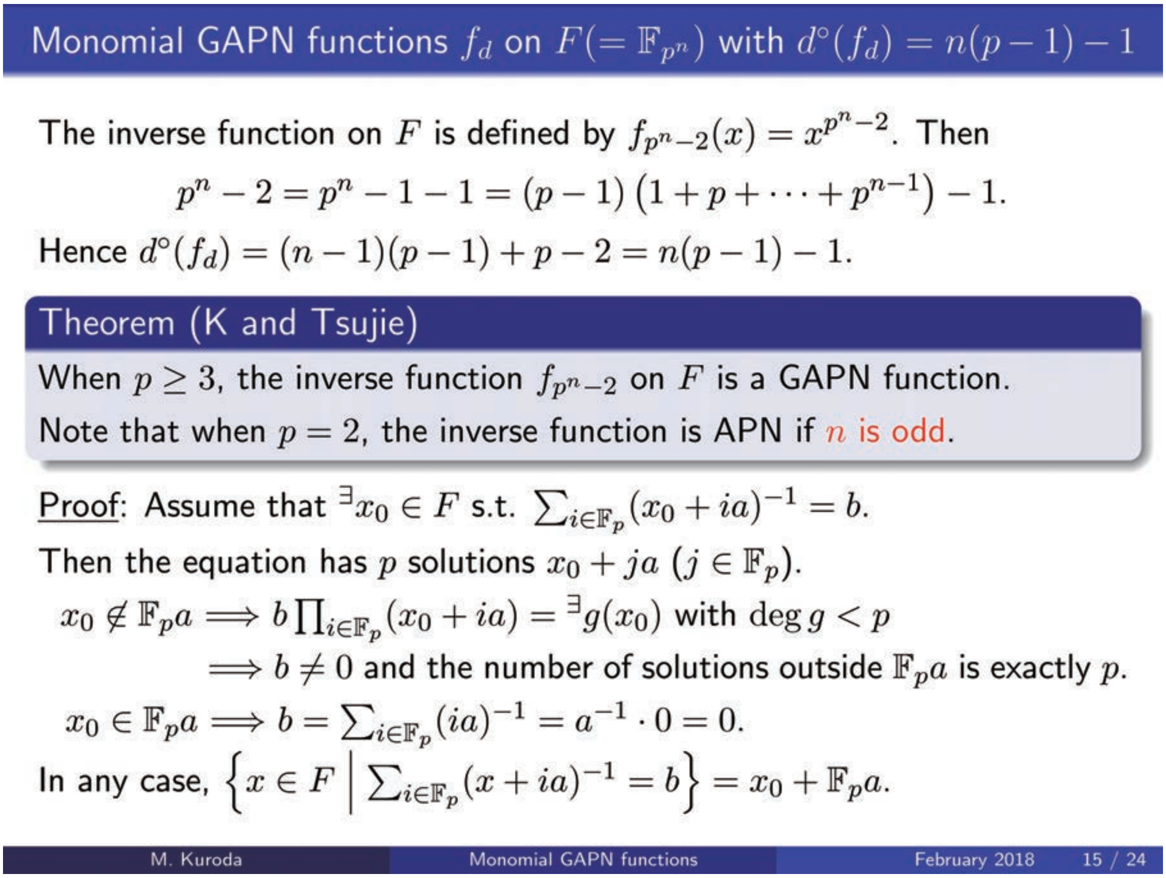
Example (a generalization of Welch function : $2^t + 3$, $n = 2t + 1$)

Let $d = p^t + p + 1$, $t = \begin{cases} \frac{n-1}{2} & (n \text{ is odd}), \\ \frac{n}{2} & (n \text{ is even}). \end{cases}$ Then

f_d is a GAPN function on $\mathbb{F}_{p^n} \iff p = 2$ and n is odd, or $p = 3$.

- When $p = 2$ and n is odd, f_d is a Welch function.
- When $p \geq 5$, f_d is not GAPN, since $d^\circ(f_d) = 3 < p$.
- When $p = 3$, we can check easily that

$$\{\beta \in \overline{\mathbb{F}_3} \mid \beta^t = -(1 + \beta)\} \cap \{\gamma \in \overline{\mathbb{F}_3} \mid \gamma^n = 1\} = \{1\}.$$



Monomial GAPN functions f_d on $F(=\mathbb{F}_{p^n})$ with $d^\circ(f_d) = n(p-1)-1$

The inverse function on F is defined by $f_{p^n-2}(x) = x^{p^n-2}$. Then

$$p^n - 2 = p^n - 1 - 1 = (p-1)\left(1+p+\cdots+p^{n-1}\right) - 1.$$

Hence $d^\circ(f_d) = (n-1)(p-1)+p-2 = n(p-1)-1$.

Theorem (K and Tsujie)

When $p \geq 3$, the inverse function f_{p^n-2} on F is a GAPN function.

Note that when $p = 2$, the inverse function is APN if n is odd.

Proof: Assume that ${}^{\exists}x_0 \in F$ s.t. $\sum_{i\in\mathbb{F}_p}(x_0+ia)^{-1} = b$.

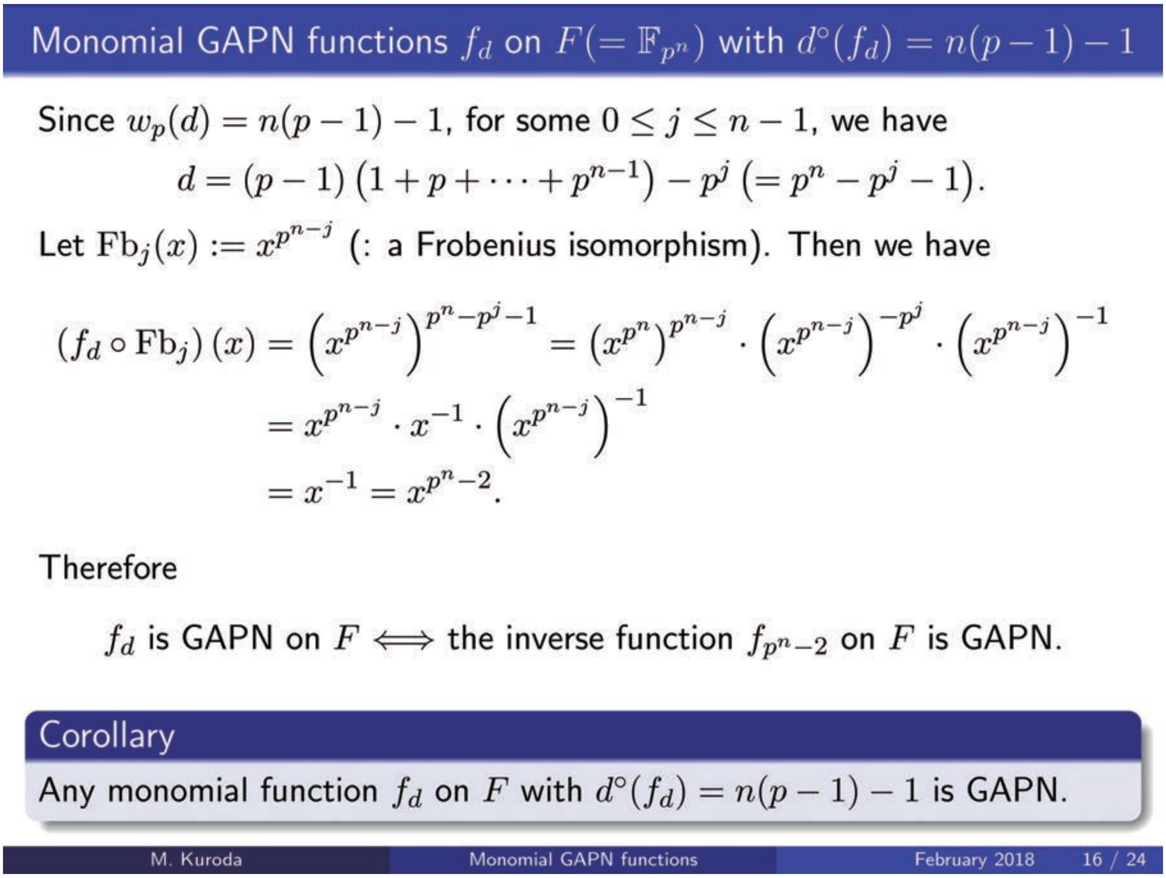
Then the equation has p solutions $x_0 + ja$ $(j \in \mathbb{F}_p)$.

$x_0 \notin \mathbb{F}_p a \Longrightarrow b\prod_{i\in\mathbb{F}_p}(x_0+ia) = {}^{\exists}g(x_0)$ with $\deg g < p$

$\Longrightarrow b \neq 0$ and the number of solutions outside $\mathbb{F}_p a$ is exactly p.

$x_0 \in \mathbb{F}_p a \Longrightarrow b = \sum_{i\in\mathbb{F}_p}(ia)^{-1} = a^{-1}\cdot 0 = 0.$

In any case, $\left\{x \in F \,\middle|\, \sum_{i\in\mathbb{F}_p}(x+ia)^{-1} = b\right\} = x_0 + \mathbb{F}_p a.$



Monomial GAPN functions f_d on $F(=\mathbb{F}_{p^n})$ with $d^\circ(f_d) = n(p-1)-1$

Since $w_p(d) = n(p-1)-1$, for some $0 \leq j \leq n-1$, we have

$$d = (p-1)\left(1+p+\cdots+p^{n-1}\right) - p^j \;(= p^n - p^j - 1).$$

Let $\mathrm{Fb}_j(x) := x^{p^{n-j}}$ (: a Frobenius isomorphism). Then we have

$$\begin{aligned}(f_d \circ \mathrm{Fb}_j)(x) &= \left(x^{p^{n-j}}\right)^{p^n-p^j-1} = \left(x^{p^n}\right)^{p^{n-j}} \cdot \left(x^{p^{n-j}}\right)^{-p^j} \cdot \left(x^{p^{n-j}}\right)^{-1} \\ &= x^{p^{n-j}} \cdot x^{-1} \cdot \left(x^{p^{n-j}}\right)^{-1} \\ &= x^{-1} = x^{p^n-2}.\end{aligned}$$

Therefore

f_d is GAPN on $F \Longleftrightarrow$ the inverse function f_{p^n-2} on F is GAPN.

Corollary

Any monomial function f_d on F with $d^\circ(f_d) = n(p-1)-1$ is GAPN.

The other monomial GAPN functions on F

Assume that $p = 3$.

- When $n \in \{1, 2, 3\}$, there are no monomial GAPN functions f_d on $\mathbb{F}_{3^n}$ with $3 < d^\circ(f_d) < 2n - 1$, clearly.
- When $n = 5$, we have the following Table:

Table 2. monomial GAPN functions f_d on $\mathbb{F}_{3^5}$ with $d^\circ(f_d) = 5$ or 7

$d^\circ(f_d)$	Exponents d
5	23, 35, 49, 73, 97, 113, 137, 169, 173, 199
7	79, 107, 197, 227

- When $n \in \{4, 6, 7, 8\}$, there are no monomial GAPN functions f_d on $\mathbb{F}_{3^n}$ with $3 < d^\circ(f_d) < 2n - 1$, by simple computations.

Conjecture 1

For sufficiently large n, there are no monomial GAPN functions f_d on $\mathbb{F}_{3^n}$ with $3 < d^\circ(f_d) < 2n - 1$.

In particular, the only 3-exceptional exponents are given by

$$d = 1 + 3^i + 3^j \quad (0 \leq i \leq j, \quad (i, j) \neq (0, 0)).$$

1. Introduction
2. Notations (algebraic degree and p-exceptional exponent)
3. A partial classification of monomial GAPN functions
4. Geometric Characterization of GAPN functions

Notations

Let $p \geq 2$, $F = \mathbb{F}_{p^n}$ and let $f_d \colon F \longrightarrow F$, $\ f_d(x) = x^d$ with $d < p^n$.

Recall that f_d is GAPN on F

$\Longleftrightarrow \#\left\{x \in F \,\middle|\, \tilde{D}_a f_d(x) = \sum_{i \in \mathbb{F}_p} (x + ia)^d = b\right\} \leq p$ for ${}^\forall a \in F^\times$, ${}^\forall b \in F$.

Definition

$$\begin{aligned} \phi_d(x, y, z) &:= \tilde{D}_{y-x} f_d(x) - \tilde{D}_{y-x} f_d(z) \\ &= \sum_{i \in \mathbb{F}_p} \left((x + i(y - x))^d - (z + i(y - x))^d \right) \in F[x, y, z], \end{aligned}$$

$$\psi_d(x, y, z) := \frac{\phi_d(x, y, z)}{(y - x)^{p-1} \prod_{j \in \mathbb{F}_p} (z - (x + j(y - x)))} \in F[x, y, z].$$

$X := V(\psi_d) \subset \mathbb{A}^3 = \operatorname{Spec}\left(\overline{\mathbb{F}_p}[x, y, z]\right)$,

$\overline{X} :=$ the projective closure of X in $\mathbb{P}^3 = \operatorname{Proj}\left(\overline{\mathbb{F}_p}[x, y, z, t]\right)$,

$H_\infty := V(t) \simeq \mathbb{P}^2 = \operatorname{Proj}\left(\overline{\mathbb{F}_p}[x, y, z]\right)$.

$\rightsquigarrow X_\infty := \overline{X} \cap H_\infty = V(\psi_d) \subset \mathbb{P}^2$ and $\overline{X} = X \cup X_\infty$.

Proposition (K)

The following are equivalent:

(i) f_d is a GAPN function on $F (= \mathbb{F}_{p^n})$.

(ii) There are no triples $(x, y, z) \in F^3$ with $x \neq y$ and $z \notin x + \mathbb{F}_p(y - x)$ s.t. $\phi_d(x, y, z) = 0$.

(iii) The affine surface $X = V(\psi_d) \subset \mathbb{A}^3$ has all its F-rational points contained in the surface $V\left((x - y) \prod_{j \in \mathbb{F}_p} (z - (x + j(y - x))) \right)$.

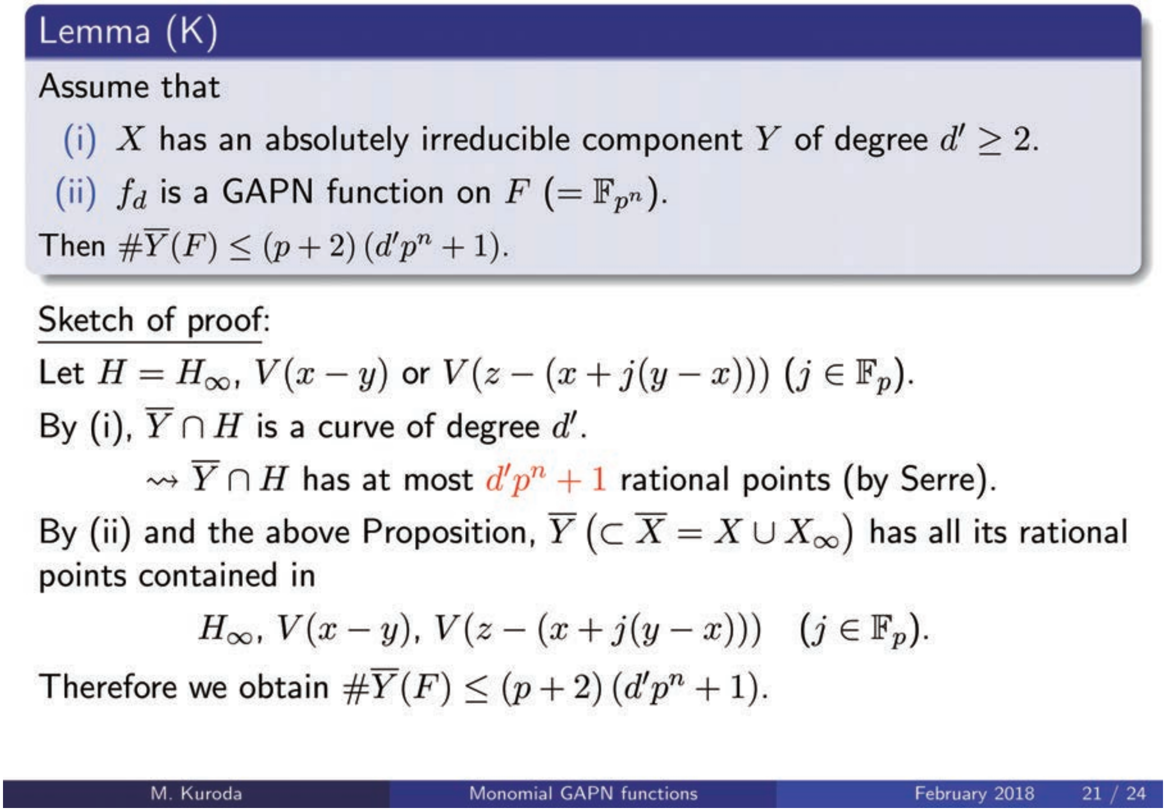
Sketch of proof: Since $\phi_d(x, y, z) = \tilde{D}_{y-x} f_d(x) - \tilde{D}_{y-x} f_d(z)$,

${}^\exists(x, y, z) \in F^3$ with $x \neq y$ and $z \notin x + \mathbb{F}_p(y - x)$ s.t. $\phi_d(x, y, z) = 0$

$\Longleftrightarrow$ the equation $\tilde{D}_{y-x} f_d(w) = \tilde{D}_{y-x} f_d(z)$ has at least $p + 1$ solutions:

$$w = x + i(y - x) \ (i \in \mathbb{F}_p) \text{ and } z \,(\notin x + \mathbb{F}_p(y - x)).$$

$\Longleftrightarrow$ f_d is not GAPN function on F.



Lemma (K)

Assume that

(i) X has an absolutely irreducible component Y of degree $d' \geq 2$.

(ii) f_d is a GAPN function on F $(= \mathbb{F}_{p^n})$.

Then $\#\overline{Y}(F) \leq (p+2)\,(d'p^n+1)$.

Sketch of proof:

Let $H = H_\infty$, $V(x-y)$ or $V(z-(x+j(y-x)))$ $(j \in \mathbb{F}_p)$.

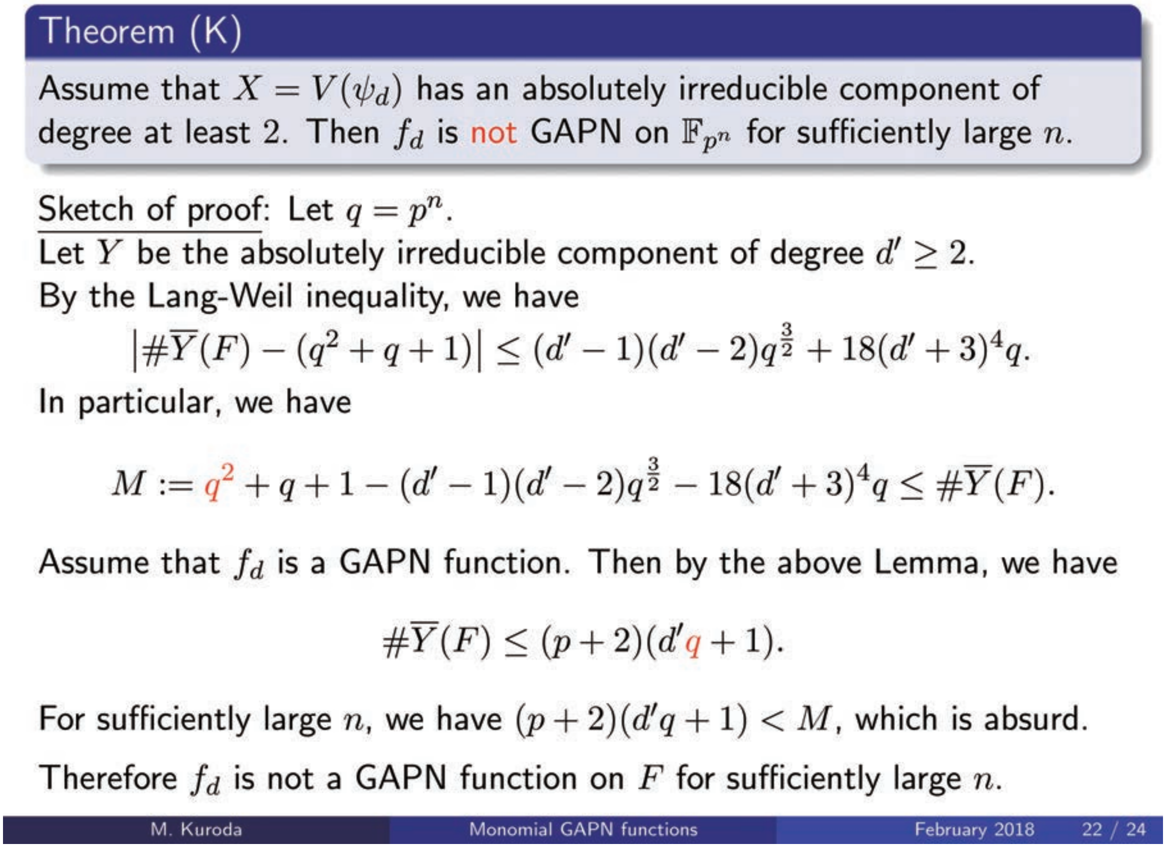
By (i), $\overline{Y} \cap H$ is a curve of degree d'.

$\rightsquigarrow \overline{Y} \cap H$ has at most $d'p^n+1$ rational points (by Serre).

By (ii) and the above Proposition, $\overline{Y}$ $\left(\subset \overline{X} = X \cup X_\infty\right)$ has all its rational points contained in

$$H_\infty,\ V(x-y),\ V(z-(x+j(y-x))) \quad (j \in \mathbb{F}_p).$$

Therefore we obtain $\#\overline{Y}(F) \leq (p+2)\,(d'p^n+1)$.



Theorem (K)

Assume that $X = V(\psi_d)$ has an absolutely irreducible component of degree at least 2. Then f_d is not GAPN on $\mathbb{F}_{p^n}$ for sufficiently large n.

Sketch of proof: Let $q = p^n$.

Let Y be the absolutely irreducible component of degree $d' \geq 2$.

By the Lang-Weil inequality, we have

$$\left|\#\overline{Y}(F) - (q^2+q+1)\right| \leq (d'-1)(d'-2)q^{\frac{3}{2}} + 18(d'+3)^4 q.$$

In particular, we have

$$M := q^2 + q + 1 - (d'-1)(d'-2)q^{\frac{3}{2}} - 18(d'+3)^4 q \leq \#\overline{Y}(F).$$

Assume that f_d is a GAPN function. Then by the above Lemma, we have

$$\#\overline{Y}(F) \leq (p+2)(d'q+1).$$

For sufficiently large n, we have $(p+2)(d'q+1) < M$, which is absurd.

Therefore f_d is not a GAPN function on F for sufficiently large n.

Monomial functions f_d with $d^\circ(f_d) > p$

Recall that $X = V(\psi_d) \subset \mathbb{A}^3 = \operatorname{Spec}\left(\overline{\mathbb{F}_p}[x,y,z]\right)$,

$\overline{X}$ = the projective closure of X in $\mathbb{P}^3 = \operatorname{Proj}\left(\overline{\mathbb{F}_p}[x,y,z,t]\right)$,

$H_\infty = V(t) \simeq \mathbb{P}^2 = \operatorname{Proj}\left(\overline{\mathbb{F}_p}[x,y,z]\right)$.

$$\rightsquigarrow X_\infty = \overline{X} \cap H_\infty = V(\psi_d) \subset \mathbb{P}^2 \text{ and } \overline{X} = X \cup X_\infty.$$

Conjecture 2

When $p = 3$, for any exponents d with $d^\circ(f_d) > 3$, the curve $X_\infty = V(\psi_d)$ $(\subset \mathbb{P}^2)$ has an absolutely irreducible component of degree at least 2.

Clearly, we have that

Conjecture 2 $\Longrightarrow$ Conjecture 1.

Conjecture 1

When $p = 3$, for sufficiently large n, there are no monomial GAPN functions f_d on $\mathbb{F}_{3^n}$ with $3 < d^\circ(f_d) < 2n - 1$. In particular, the only 3-exceptional exponents are given by $d = 1 + 3^i + 3^j \quad (0 \le i \le j, \quad (i,j) \neq (0,0))$.

When $p = 2$

The following conjecture was proved by Hernando and McGuire (2011):

Conjecture

When $p = 2$, the curve $X_\infty = V(\psi_d)$ $(\subset \mathbb{P}^2)$ has an absolutely irreducible component of degree at least 2 for all d not of the form $2^i + 1$ (Gold) or $2^{2i} - 2^i + 1$ (Kasami).

By the above theorem, it is clear that

Theorem (Hernando and McGuire, 2011)

The only 2-exceptional exponents are Gold numbers $(2^i + 1)$ and Kasami numbers $(2^{2i} - 2^i + 1)$.

Elliptic Curve Method with Complex Multiplication Method

Yusuke AIKAWA

Hokkaido University
AIST

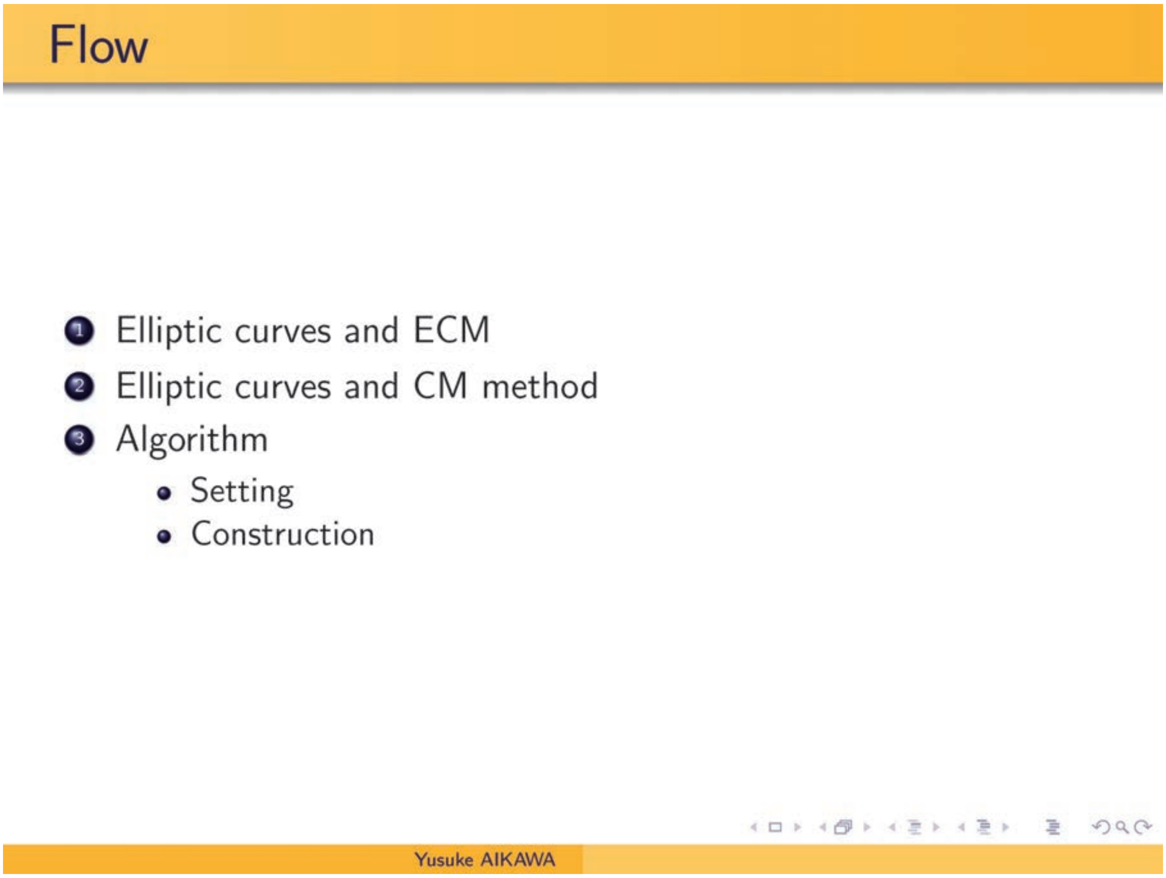
A joint work with K.Nuida(AIST/JST PRESTO) and
M.Shirase(Future University Hakodate)

2. 6. 2018

Workshop on analysis of mathematical cryptography
via algebraic methods

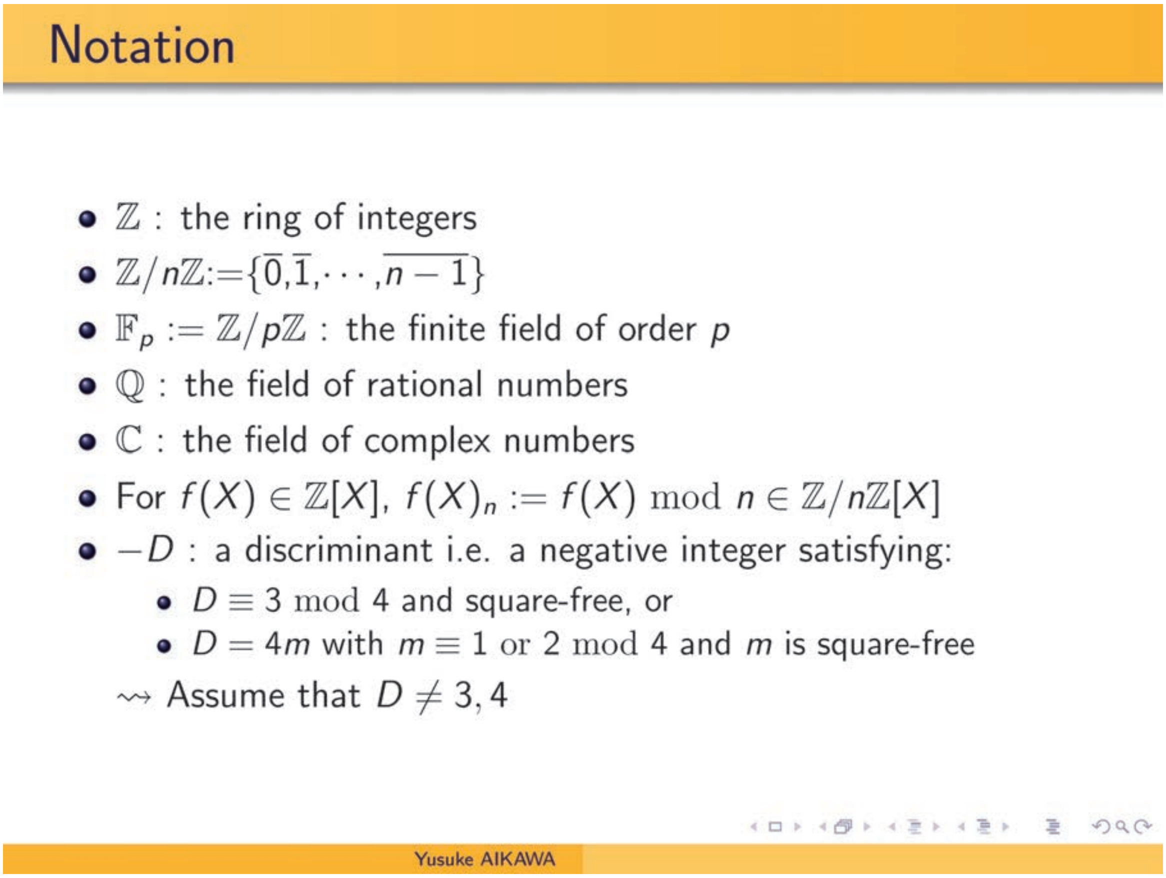
Aim of the Talk

1. Present a new factoring algorithm
   ⇝ A combination of the Elliptic Curve Method and the Complex Multiplication Method
   - Input date : a discriminant $-D \in \mathbb{Z}$ and its class polynomial $H_{-D}(X)$
   - Works in polynomial time for composites having a prime factor of the special form:
     - $4p = 1 + Dv^2 (v \in \mathbb{Z})$
     - $4p = t^2 + Dv^2 (v \in \mathbb{Z})$ and $p + 1 - t$ is a smooth integer
2. Give an introductory explanation of key tools
   - The elliptic curves and the Elliptic Curve Method (ECM)
   - The Complex Multiplication Method (CM method)



Flow

1. Elliptic curves and ECM
2. Elliptic curves and CM method
3. Algorithm
   - Setting
   - Construction



Notation

- $\mathbb{Z}$: the ring of integers
- $\mathbb{Z}/n\mathbb{Z}$:=$\{\overline{0},\overline{1},\cdots,\overline{n-1}\}$
- $\mathbb{F}_p := \mathbb{Z}/p\mathbb{Z}$: the finite field of order p
- $\mathbb{Q}$: the field of rational numbers
- $\mathbb{C}$: the field of complex numbers
- For $f(X) \in \mathbb{Z}[X]$, $f(X)_n := f(X) \bmod n \in \mathbb{Z}/n\mathbb{Z}[X]$
- $-D$: a discriminant i.e. a negative integer satisfying:
  - $D \equiv 3 \bmod 4$ and square-free, or
  - $D = 4m$ with $m \equiv 1 \text{ or } 2 \bmod 4$ and m is square-free

$\rightsquigarrow$ Assume that $D \neq 3, 4$

Elliptic Curves

Let K be a field with ch$K \neq 2, 3$.
An elliptic curves over K is an algebraic curves defined by the so-called Weierstrass equation:

$$E : Y^2 = X^3 + AX + B \quad (A, B \in K \text{ and } 4A^3 + 27B^2 \neq 0).$$

Define the set of rational points of E on a field K:

$$E(K) := \left\{(x, y) \in K \times K | y^2 = x^3 + Ax + B\right\} \cup \left\{\infty\right\}$$

where $\infty = [0 : 1 : 0] \in \mathbb{P}^2$.
$\rightsquigarrow$ $E(K)$ carries a group structure with a unit ∞ (the Mordell-Weil group of E).

Addition Formula for the Weierstrass Equation

For $P_1 = (x_1, y_1), P_2 = (x_2, y_2)$ with $x_1 \neq x_2$ and $P_1, P_2 \neq \infty$, define $P_3 := P_1 + P_2 = (x_3, y_3)$ as follows:

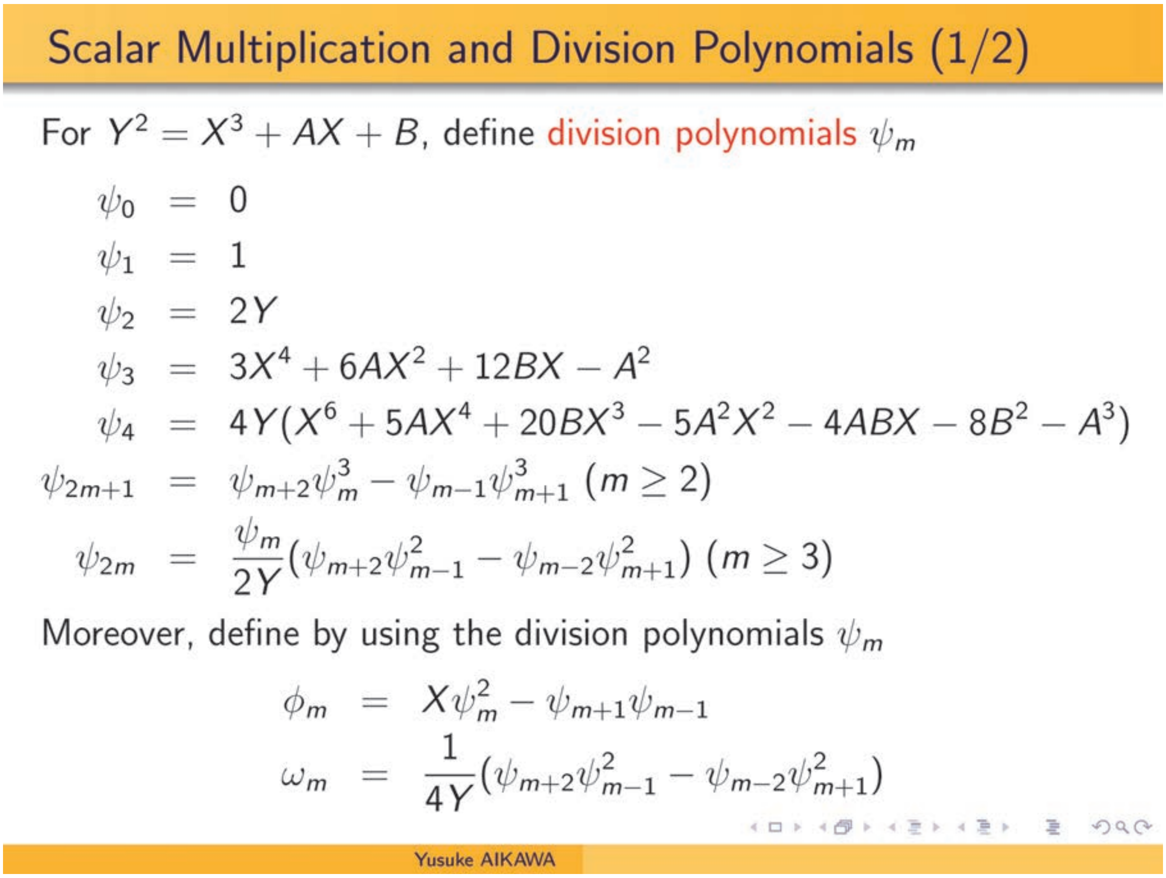
$$x_3 = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)^2 - x_1 - x_2,$$

$$y_3 = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)(x_1 - x_3) - y_1.$$

In the context of cryptography, the scalar multiplication

$$nP = \overbrace{P + P + \cdots + P}^{n}$$

is important.
$\rightsquigarrow$ Division polynomials.



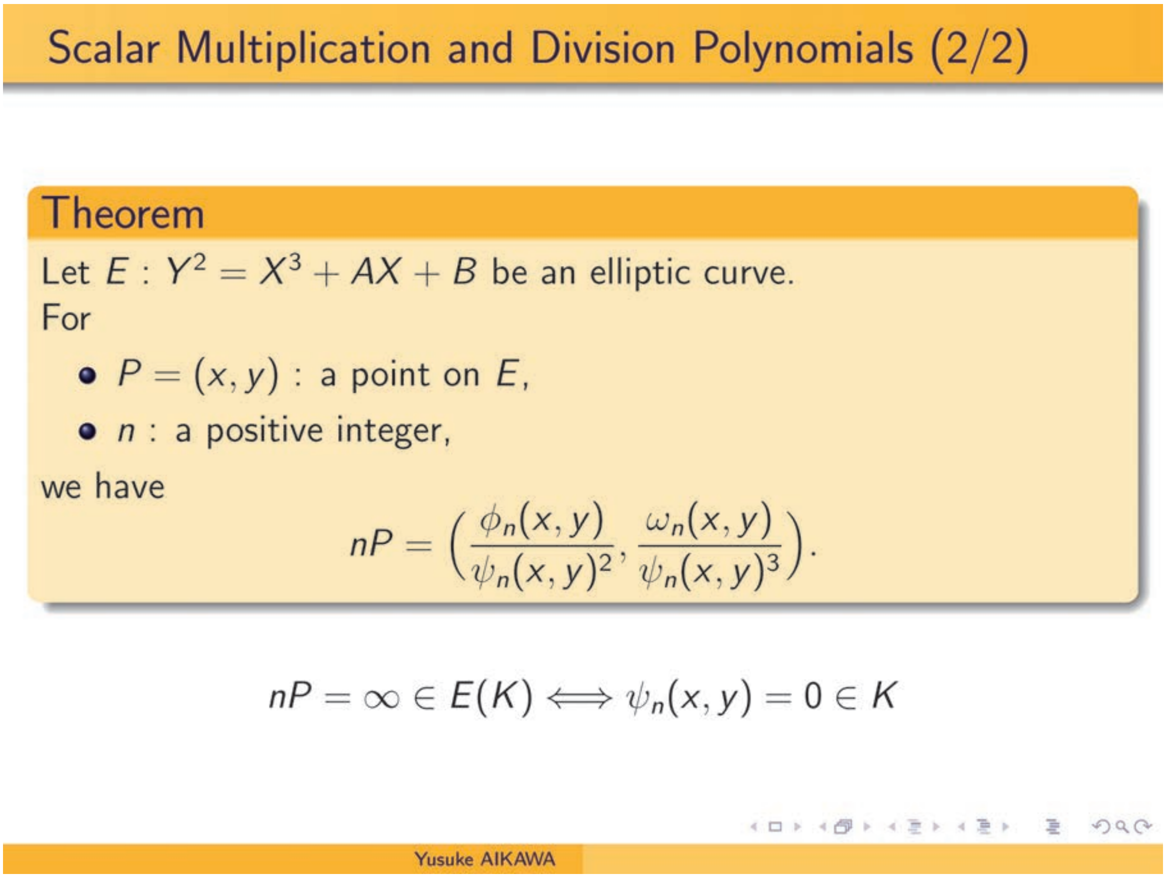
Scalar Multiplication and Division Polynomials (1/2)

For $Y^2 = X^3 + AX + B$, define division polynomials ψ_m

$$
\begin{aligned}
\psi_0 &= 0 \\
\psi_1 &= 1 \\
\psi_2 &= 2Y \\
\psi_3 &= 3X^4 + 6AX^2 + 12BX - A^2 \\
\psi_4 &= 4Y(X^6 + 5AX^4 + 20BX^3 - 5A^2X^2 - 4ABX - 8B^2 - A^3) \\
\psi_{2m+1} &= \psi_{m+2}\psi_m^3 - \psi_{m-1}\psi_{m+1}^3 \ (m \geq 2) \\
\psi_{2m} &= \frac{\psi_m}{2Y}(\psi_{m+2}\psi_{m-1}^2 - \psi_{m-2}\psi_{m+1}^2) \ (m \geq 3)
\end{aligned}
$$

Moreover, define by using the division polynomials ψ_m

$$
\begin{aligned}
\phi_m &= X\psi_m^2 - \psi_{m+1}\psi_{m-1} \\
\omega_m &= \frac{1}{4Y}(\psi_{m+2}\psi_{m-1}^2 - \psi_{m-2}\psi_{m+1}^2)
\end{aligned}
$$



Scalar Multiplication and Division Polynomials (2/2)

Theorem

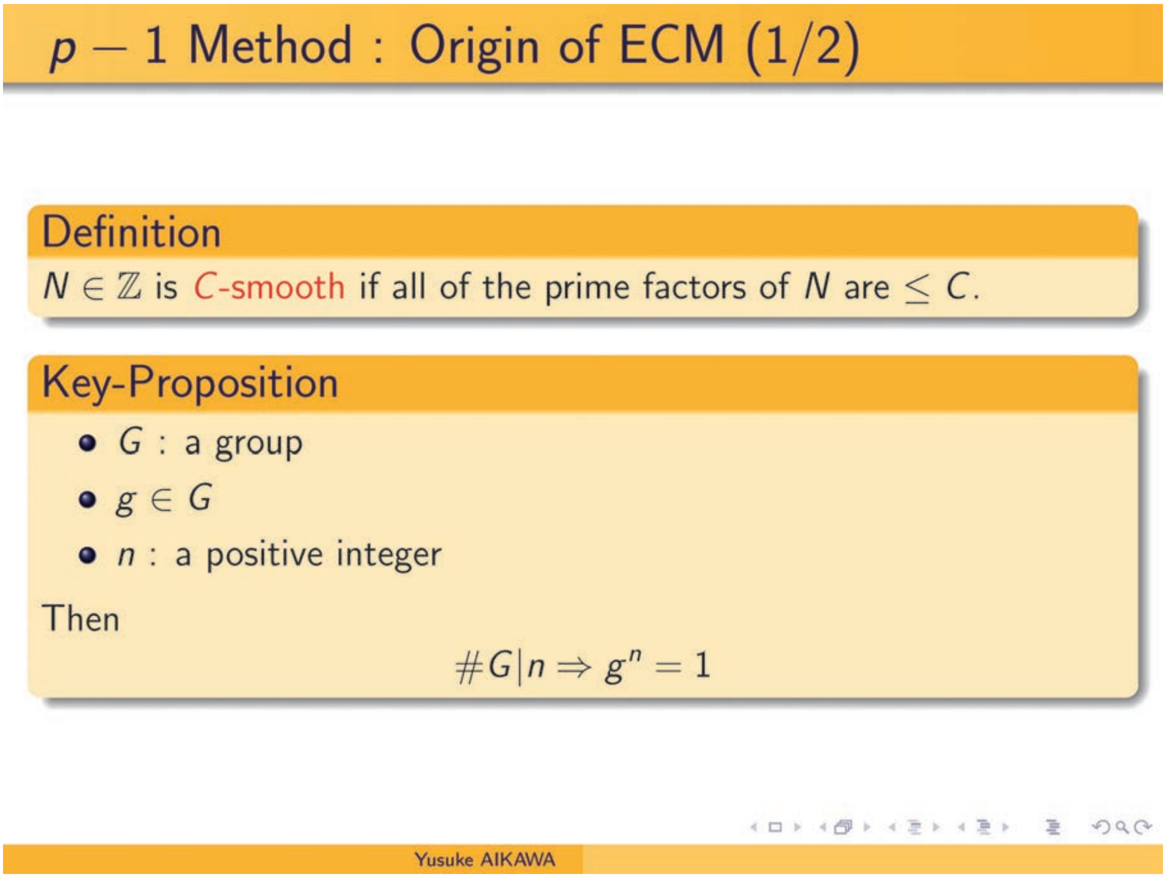
Let $E : Y^2 = X^3 + AX + B$ be an elliptic curve.
For

- $P = (x, y)$: a point on E,
- n : a positive integer,

we have

$$nP = \left(\frac{\phi_n(x,y)}{\psi_n(x,y)^2}, \frac{\omega_n(x,y)}{\psi_n(x,y)^3}\right).$$

$$nP = \infty \in E(K) \Longleftrightarrow \psi_n(x,y) = 0 \in K$$



$p-1$ Method : Origin of ECM (1/2)

Definition

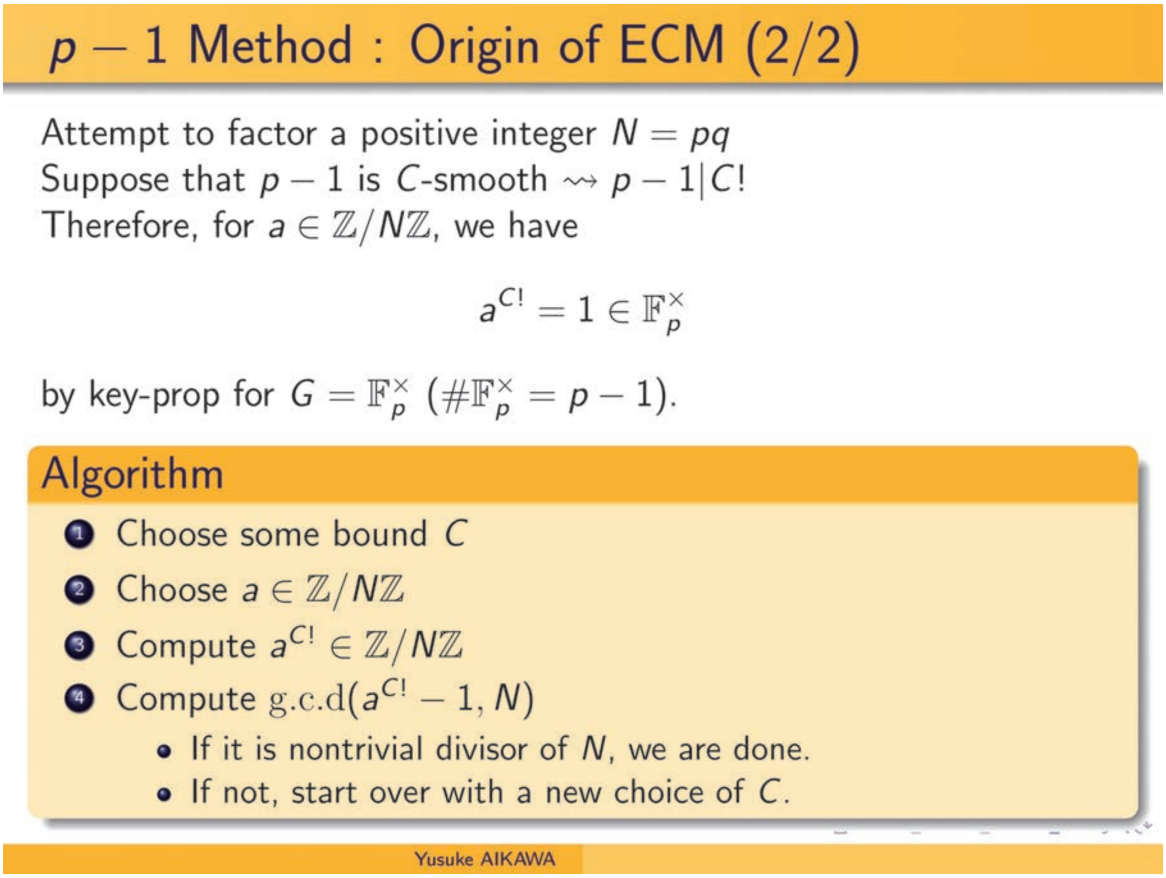
$N \in \mathbb{Z}$ is C-smooth if all of the prime factors of N are $\leq C$.

Key-Proposition

- G : a group
- $g \in G$
- n : a positive integer

Then

$$\#G | n \Rightarrow g^n = 1$$



$p-1$ Method : Origin of ECM (2/2)

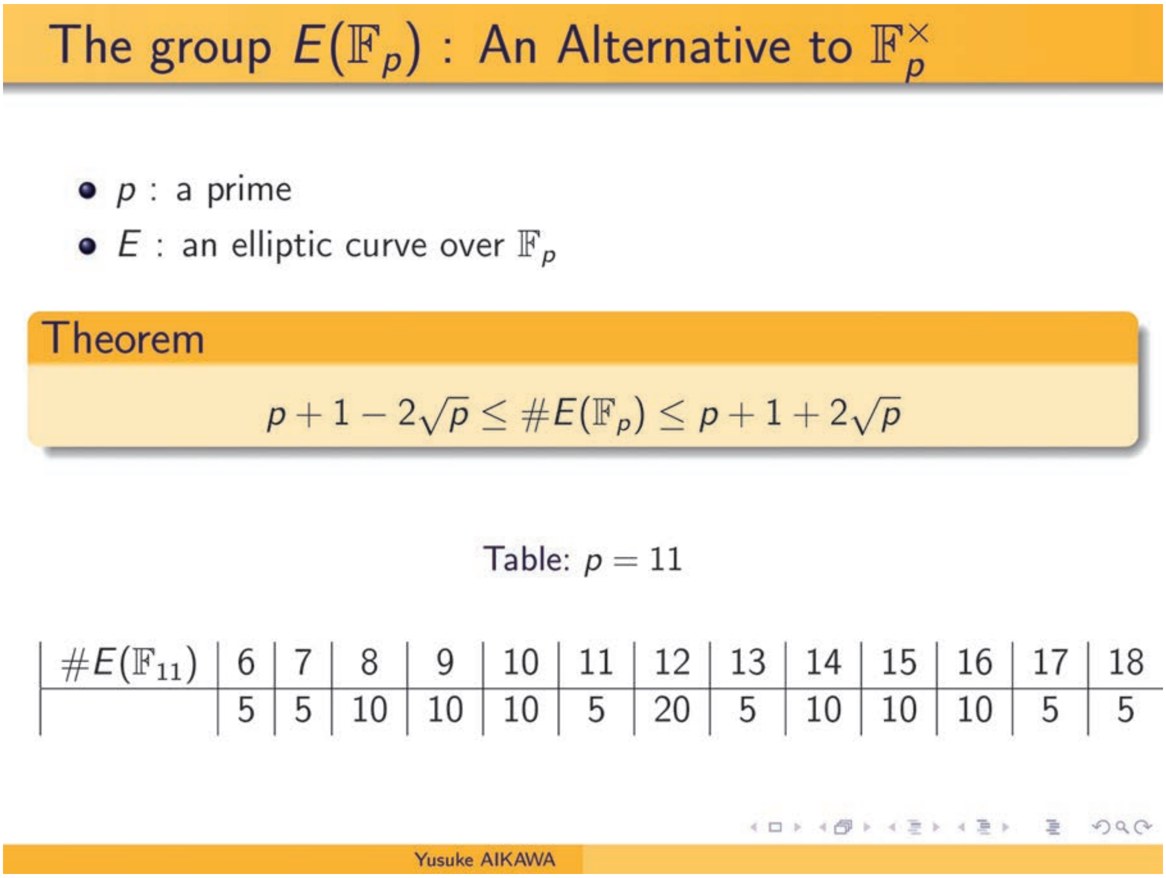
Attempt to factor a positive integer $N = pq$
Suppose that $p-1$ is C-smooth $\rightsquigarrow p-1 | C!$
Therefore, for $a \in \mathbb{Z}/N\mathbb{Z}$, we have

$$a^{C!} = 1 \in \mathbb{F}_p^{\times}$$

by key-prop for $G = \mathbb{F}_p^{\times}$ $(\#\mathbb{F}_p^{\times} = p-1)$.

Algorithm

1. Choose some bound C
2. Choose $a \in \mathbb{Z}/N\mathbb{Z}$
3. Compute $a^{C!} \in \mathbb{Z}/N\mathbb{Z}$
4. Compute $\mathrm{g.c.d}(a^{C!} - 1, N)$
    - If it is nontrivial divisor of N, we are done.
    - If not, start over with a new choice of C.



The group $E(\mathbb{F}_p)$: An Alternative to $\mathbb{F}_p^\times$

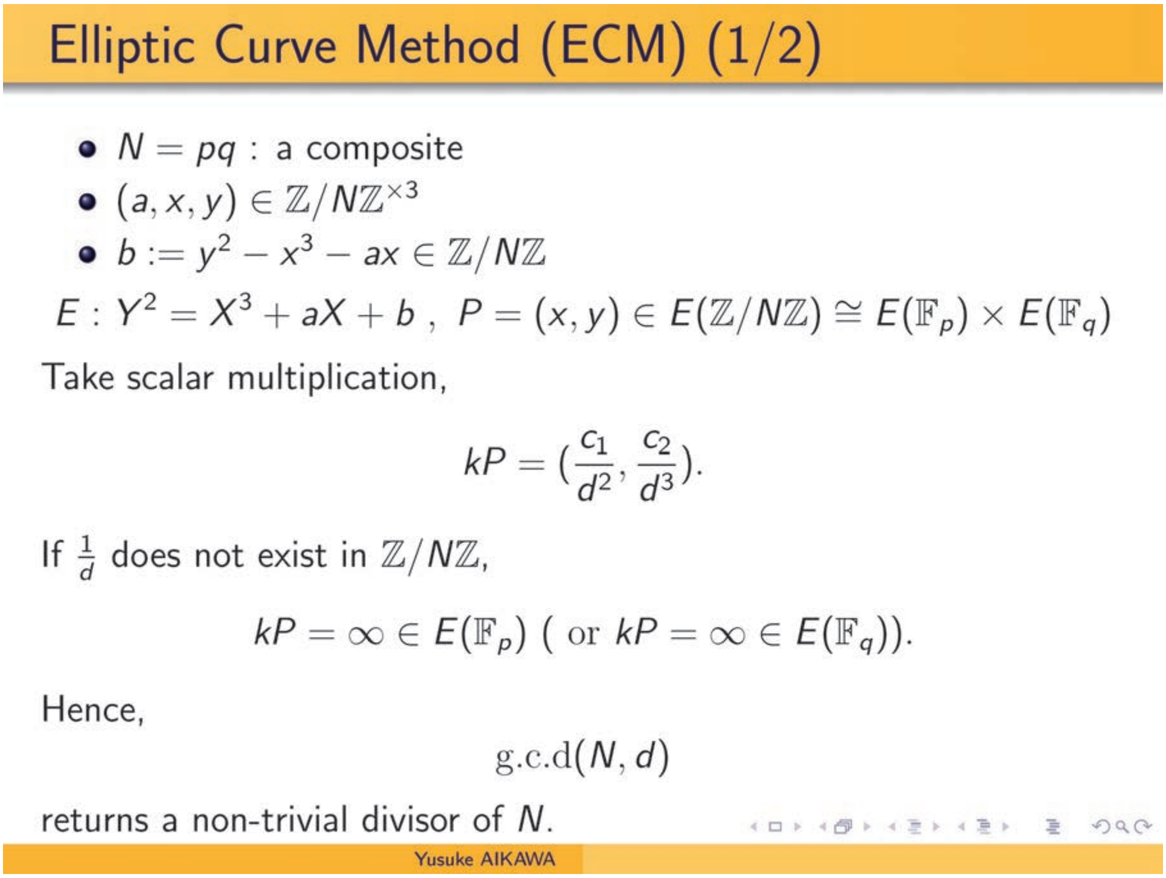
- p : a prime
- E : an elliptic curve over $\mathbb{F}_p$

Theorem

$$p+1-2\sqrt{p} \leq \#E(\mathbb{F}_p) \leq p+1+2\sqrt{p}$$

Table: $p = 11$

$\#E(\mathbb{F}_{11})$	6	7	8	9	10	11	12	13	14	15	16	17	18
	5	5	10	10	10	5	20	5	10	10	10	5	5



Elliptic Curve Method (ECM) (1/2)

- $N = pq$: a composite
- $(a, x, y) \in \mathbb{Z}/N\mathbb{Z}^{\times 3}$
- $b := y^2 - x^3 - ax \in \mathbb{Z}/N\mathbb{Z}$

$E : Y^2 = X^3 + aX + b$, $P = (x, y) \in E(\mathbb{Z}/N\mathbb{Z}) \cong E(\mathbb{F}_p) \times E(\mathbb{F}_q)$

Take scalar multiplication,

$$kP = (\frac{c_1}{d^2}, \frac{c_2}{d^3}).$$

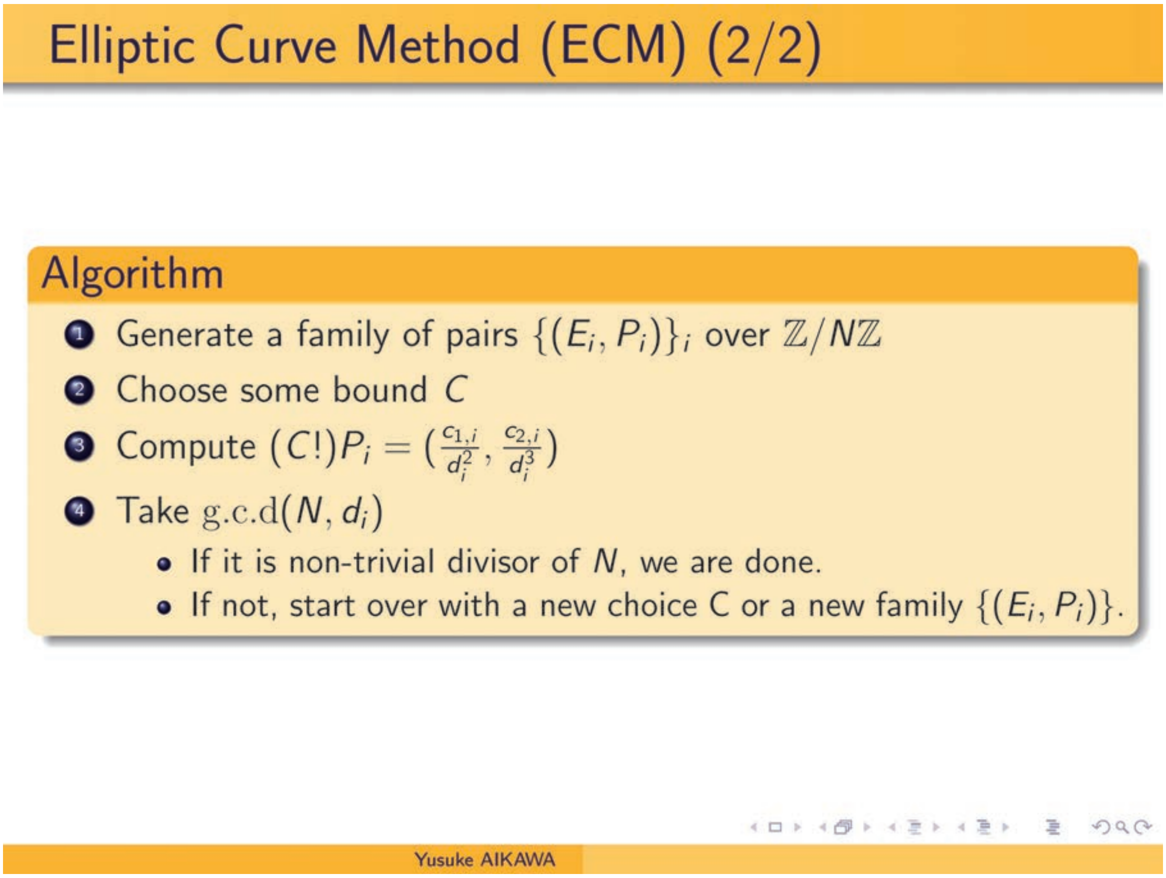
If $\frac{1}{d}$ does not exist in $\mathbb{Z}/N\mathbb{Z}$,

$$kP = \infty \in E(\mathbb{F}_p) \text{ (or } kP = \infty \in E(\mathbb{F}_q)).$$

Hence,

$$\text{g.c.d}(N, d)$$

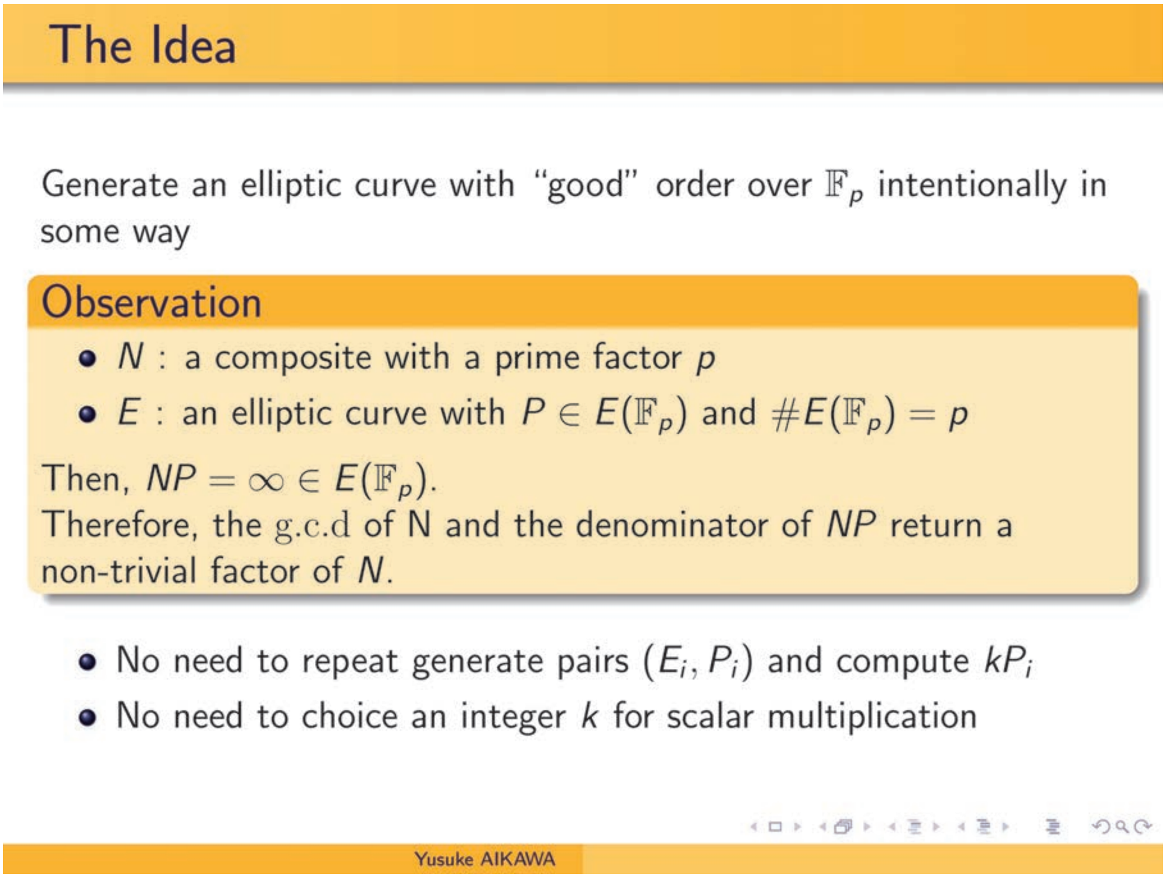
returns a non-trivial divisor of N.



Elliptic Curve Method (ECM) (2/2)

Algorithm

1. Generate a family of pairs $\{(E_i, P_i)\}_i$ over $\mathbb{Z}/N\mathbb{Z}$
2. Choose some bound C
3. Compute $(C!)P_i = (\frac{c_{1,i}}{d_i^2}, \frac{c_{2,i}}{d_i^3})$
4. Take $\mathrm{g.c.d}(N, d_i)$
    - If it is non-trivial divisor of N, we are done.
    - If not, start over with a new choice C or a new family $\{(E_i, P_i)\}$.



The Idea

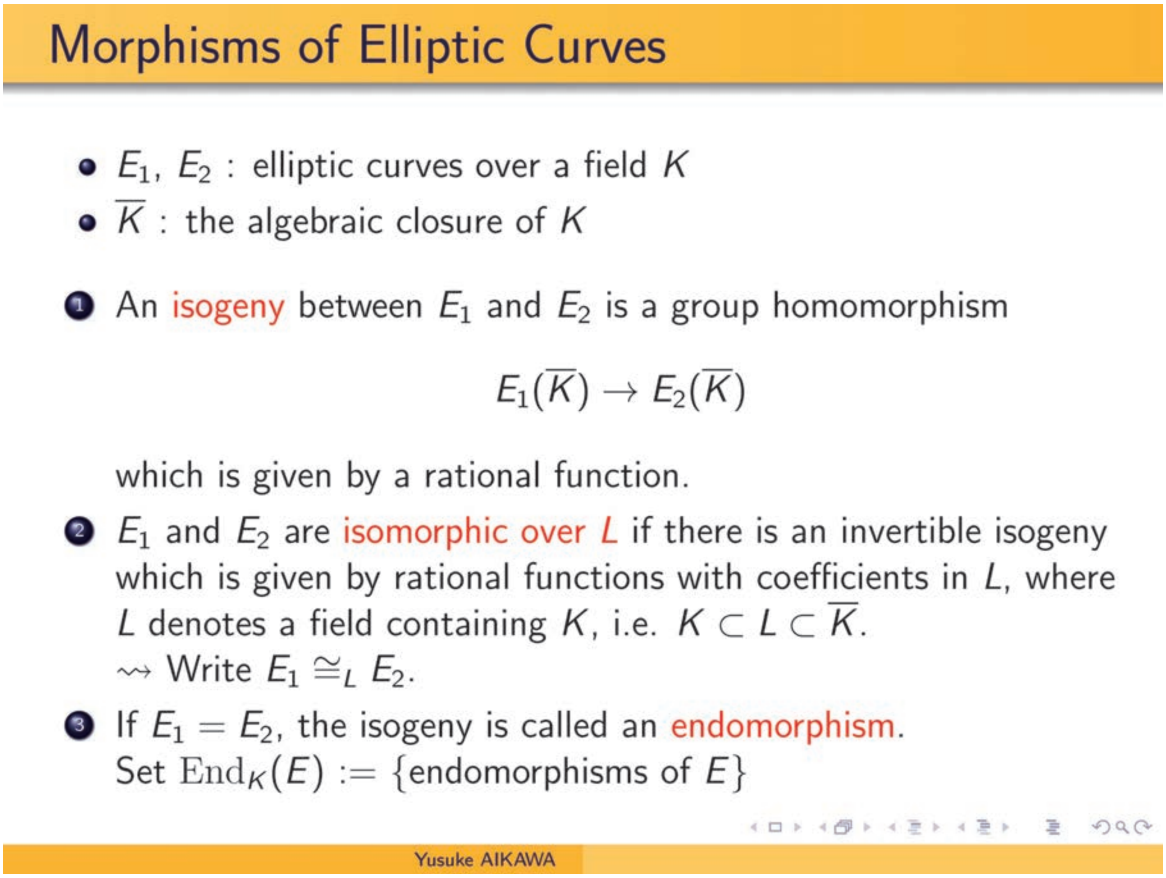
Generate an elliptic curve with "good" order over $\mathbb{F}_p$ intentionally in some way

Observation

- N : a composite with a prime factor p
- E : an elliptic curve with $P \in E(\mathbb{F}_p)$ and $\#E(\mathbb{F}_p) = p$

Then, $NP = \infty \in E(\mathbb{F}_p)$.
Therefore, the $\mathrm{g.c.d}$ of N and the denominator of NP return a non-trivial factor of N.

- No need to repeat generate pairs (E_i, P_i) and compute kP_i
- No need to choice an integer k for scalar multiplication

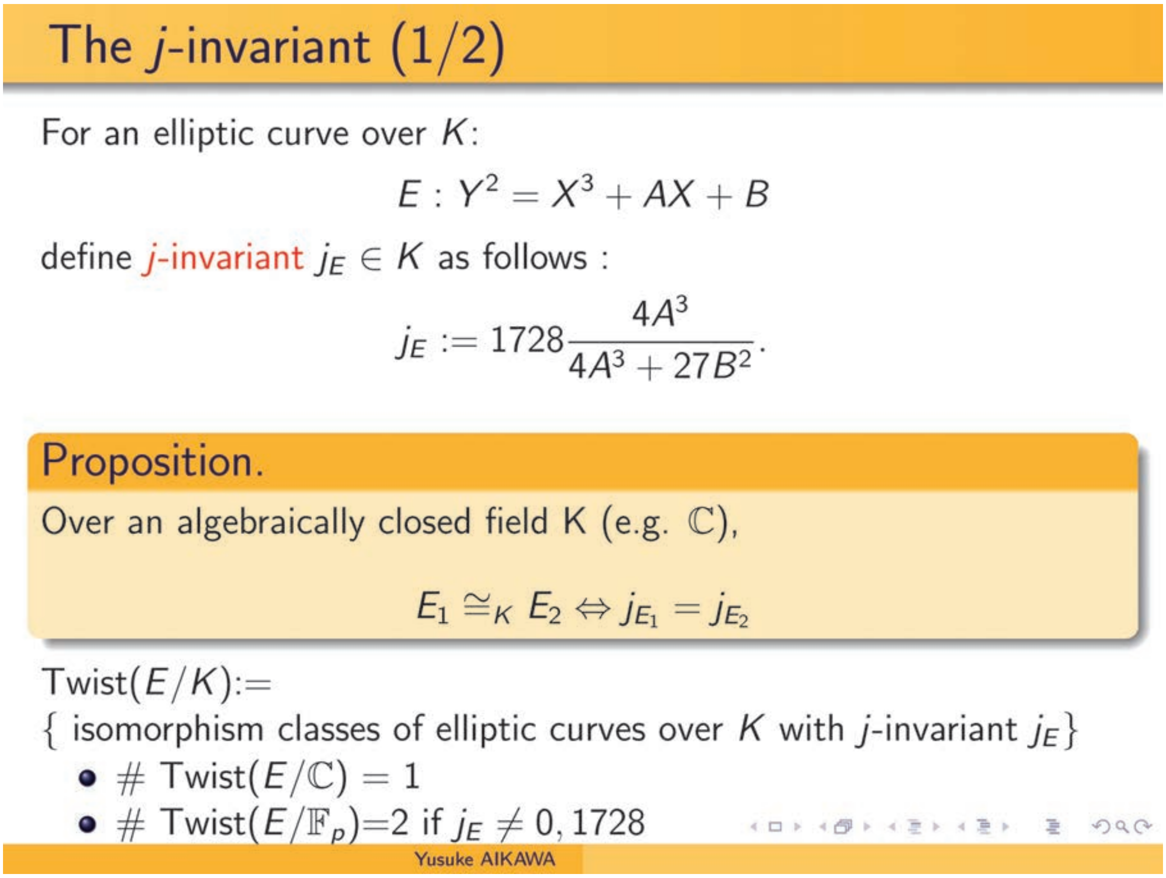


Morphisms of Elliptic Curves

- E_1, E_2 : elliptic curves over a field K
- $\overline{K}$: the algebraic closure of K

1. An isogeny between E_1 and E_2 is a group homomorphism
$$E_1(\overline{K}) \to E_2(\overline{K})$$
which is given by a rational function.
2. E_1 and E_2 are isomorphic over L if there is an invertible isogeny which is given by rational functions with coefficients in L, where L denotes a field containing K, i.e. $K \subset L \subset \overline{K}$.
$\rightsquigarrow$ Write $E_1 \cong_L E_2$.
3. If $E_1 = E_2$, the isogeny is called an endomorphism.
Set $\mathrm{End}_K(E) := \{$endomorphisms of $E\}$

Yusuke AIKAWA



The j-invariant (1/2)

For an elliptic curve over K:
$$E : Y^2 = X^3 + AX + B$$
define j-invariant $j_E \in K$ as follows :
$$j_E := 1728\frac{4A^3}{4A^3 + 27B^2}.$$

Proposition.

Over an algebraically closed field K (e.g. $\mathbb{C}$),
$$E_1 \cong_K E_2 \Leftrightarrow j_{E_1} = j_{E_2}$$

Twist(E/K):=
$\{$ isomorphism classes of elliptic curves over K with j-invariant $j_E\}$

- $\#$ Twist$(E/\mathbb{C}) = 1$
- $\#$ Twist$(E/\mathbb{F}_p)$=2 if $j_E \neq 0, 1728$

Yusuke AIKAWA

The j-invariant (2/2)

For $j_0 \neq 0, 1728 \in K$,

$$E_{j_0} : Y^2 = X^3 + \frac{3j_0}{1728 - j_0}X + \frac{2j_0}{1728 - j_0}$$

has j-invariant j_0.
Over $\mathbb{F}_p$, for $c \in \mathbb{F}_p$ quadratic non-residue,

$$E_{j_0}^c : Y^2 = X^3 + \frac{3c^2 j_0}{1728 - j_0}X + \frac{2c^3 j_0}{1728 - j_0}$$

has same j-invariant as E_{j_0}, but not isomorphic to E_{j_0} over $\mathbb{F}_p$.

The Ring $End_{\mathbb{C}}(E)$ and Complex Multiplication

The endomorphisms $[n] : E(\overline{K}) \to E(\overline{K})$; $P \mapsto nP$ yield

$$\mathbb{Z} \hookrightarrow \mathrm{End}(E); n \mapsto [n]$$

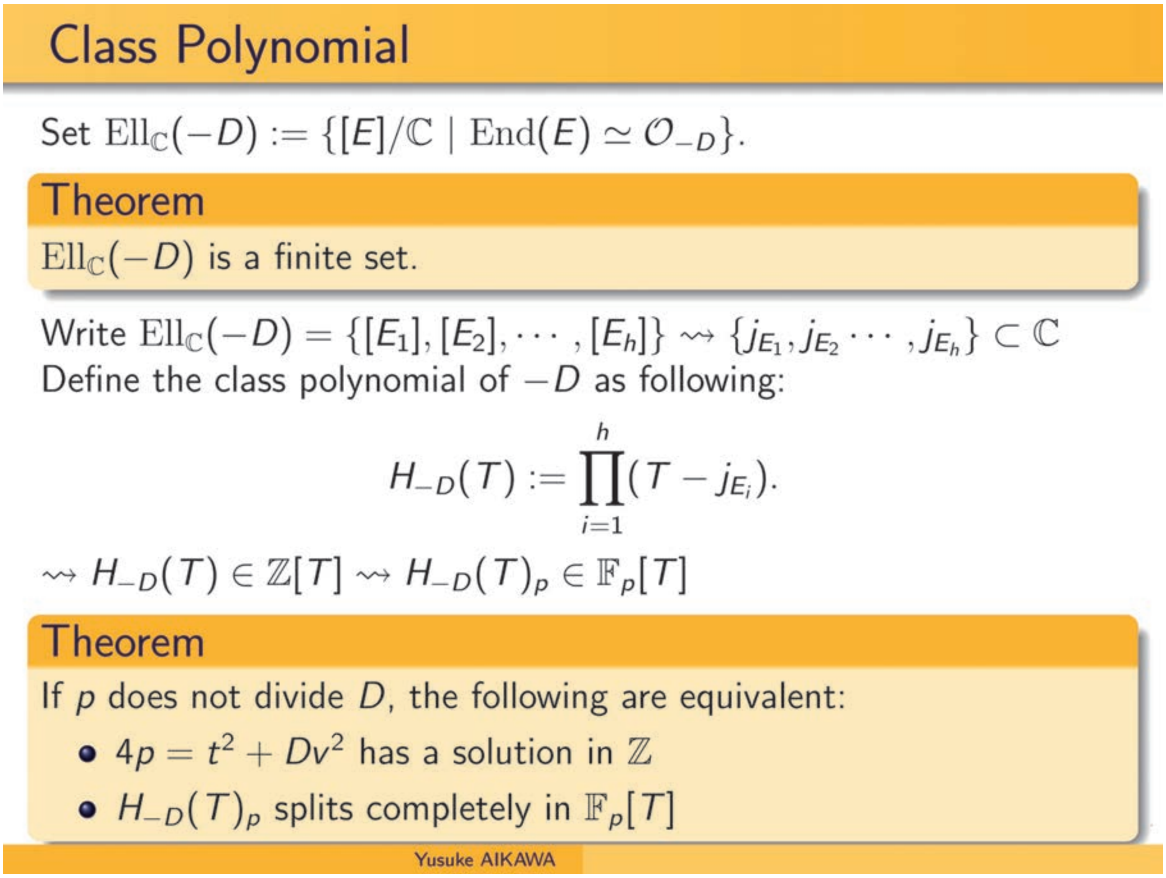
- $-D$: a discriminant
- $\mathbb{Q}(\sqrt{-D}) := \{a + \sqrt{-D}b \in \mathbb{C} | a, b \in \mathbb{Q}\}$: an imaginary quadratic field
- $\mathcal{O}_{-D} := \mathbb{Z}[\frac{-D+\sqrt{-D}}{2}]$: the maximal order of $\mathbb{Q}(\sqrt{-D})$

Theorem and Definition

Let E be an elliptic curve over $\mathbb{C}$. Then, $\mathrm{End}_{\mathbb{C}}(E)$ is isomorphic to ;

- $\mathbb{Z}$ or,
- an order of some imaginaly quadratic field $\mathbb{Q}(\sqrt{-D})$.

E has complex multiplication (CM) if $\mathrm{End}(E) \neq \mathbb{Z}$.
$\rightsquigarrow \mathrm{End}_{\mathbb{C}}(E) \cong \mathcal{O}_{-D}$



Class Polynomial

Set $\mathrm{Ell}_{\mathbb{C}}(-D) := \{[E]/\mathbb{C} \mid \mathrm{End}(E) \simeq \mathcal{O}_{-D}\}$.

Theorem

$\mathrm{Ell}_{\mathbb{C}}(-D)$ is a finite set.

Write $\mathrm{Ell}_{\mathbb{C}}(-D) = \{[E_1], [E_2], \cdots, [E_h]\} \rightsquigarrow \{j_{E_1}, j_{E_2} \cdots, j_{E_h}\} \subset \mathbb{C}$
Define the class polynomial of $-D$ as following:

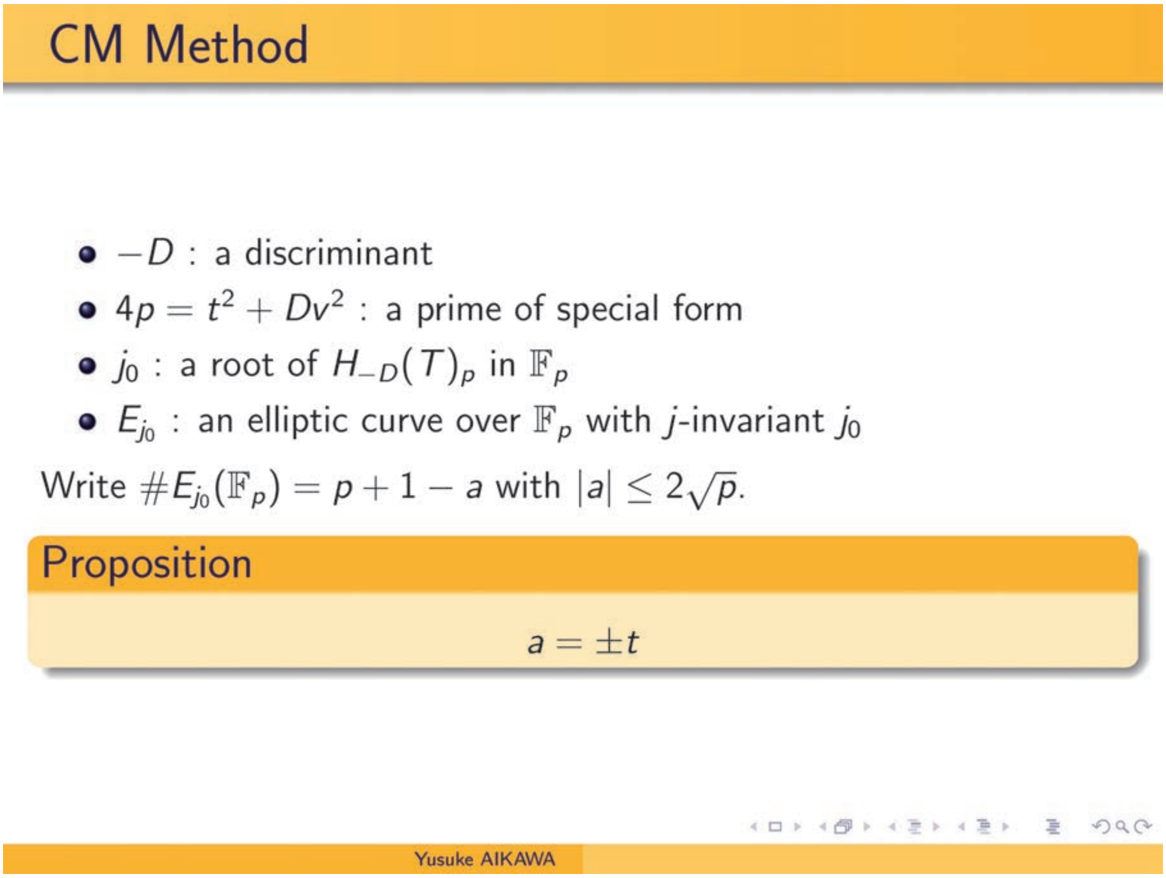
$$H_{-D}(T) := \prod_{i=1}^{h} (T - j_{E_i}).$$

$\rightsquigarrow H_{-D}(T) \in \mathbb{Z}[T] \rightsquigarrow H_{-D}(T)_p \in \mathbb{F}_p[T]$

Theorem

If p does not divide D, the following are equivalent:

- $4p = t^2 + Dv^2$ has a solution in $\mathbb{Z}$
- $H_{-D}(T)_p$ splits completely in $\mathbb{F}_p[T]$



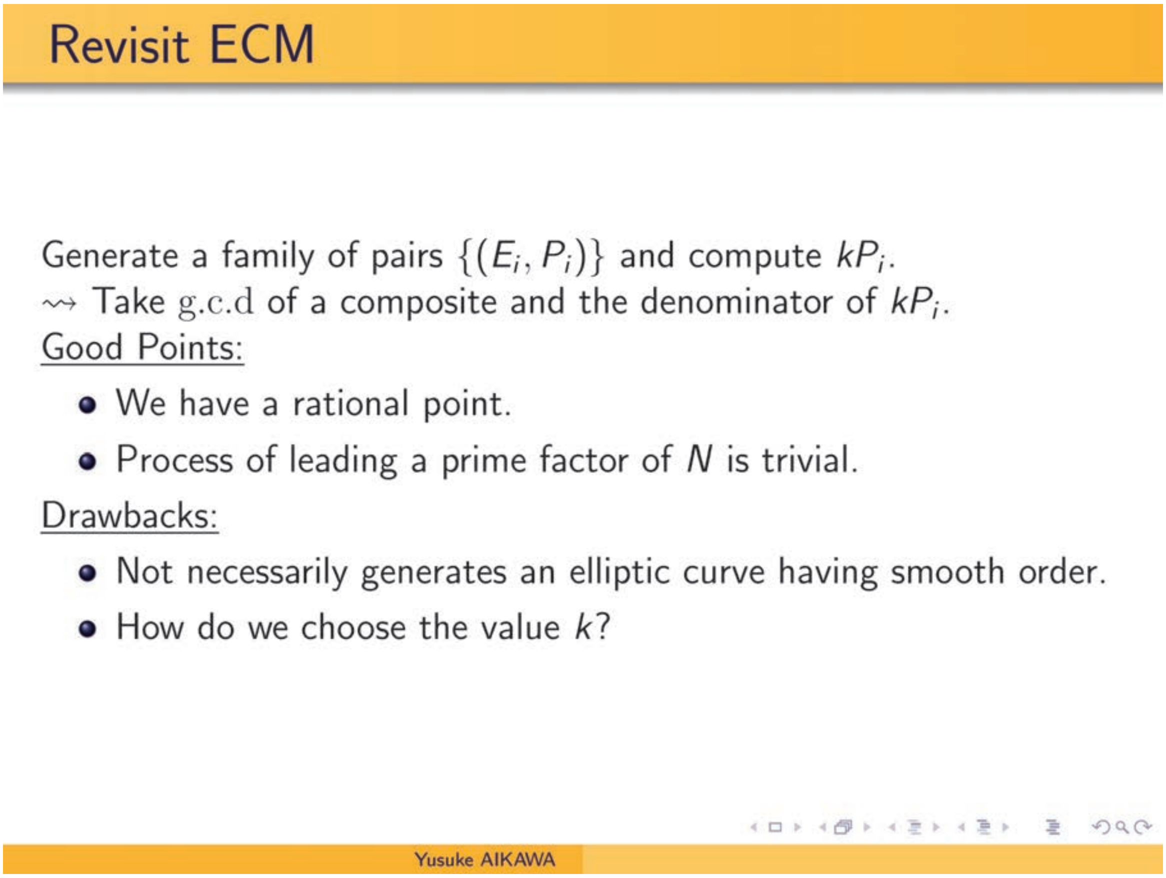
CM Method

- $-D$: a discriminant
- $4p = t^2 + Dv^2$: a prime of special form
- j_0 : a root of $H_{-D}(T)_p$ in $\mathbb{F}_p$
- E_{j_0} : an elliptic curve over $\mathbb{F}_p$ with j-invariant j_0

Write $\#E_{j_0}(\mathbb{F}_p) = p + 1 - a$ with $|a| \leq 2\sqrt{p}$.

Proposition

$$a = \pm t$$



Revisit ECM

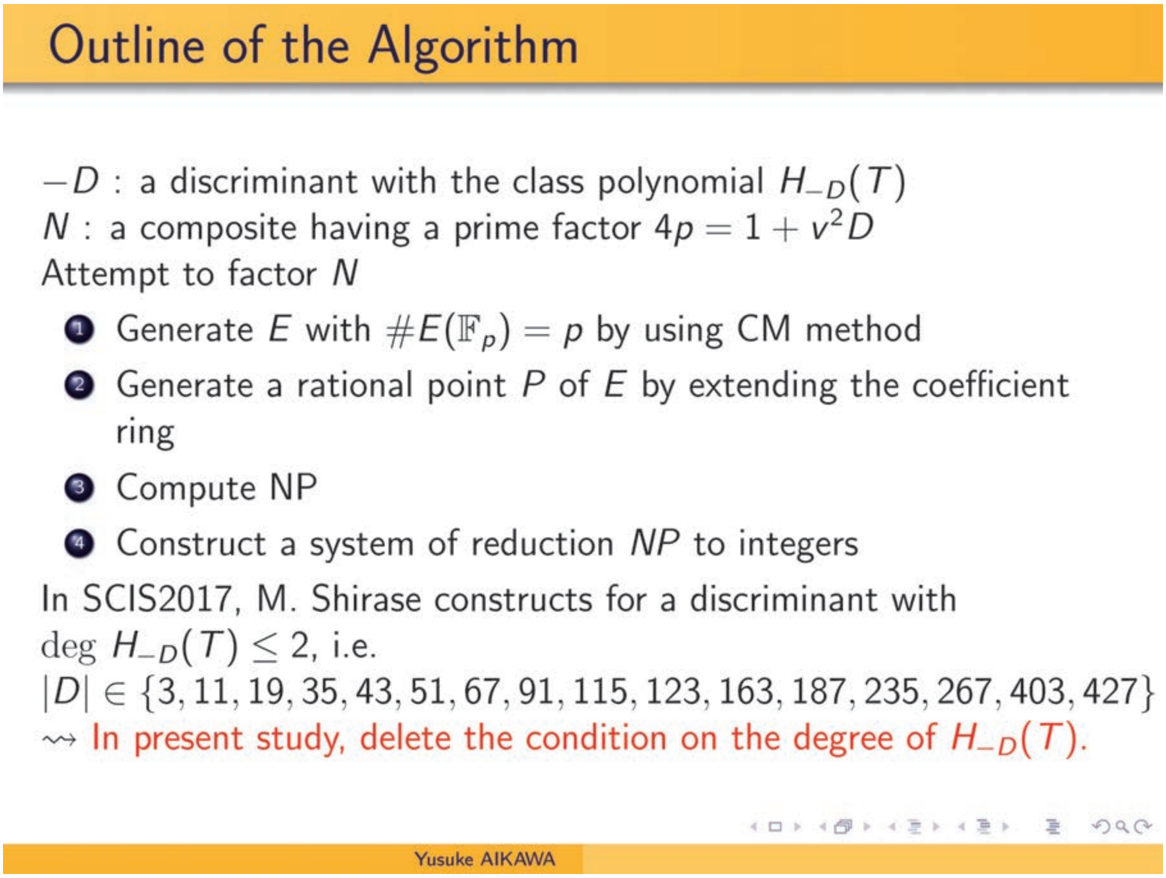
Generate a family of pairs $\{(E_i, P_i)\}$ and compute kP_i.
$\rightsquigarrow$ Take $\mathrm{g.c.d}$ of a composite and the denominator of kP_i.

Good Points:

- We have a rational point.
- Process of leading a prime factor of N is trivial.

Drawbacks:

- Not necessarily generates an elliptic curve having smooth order.
- How do we choose the value k?

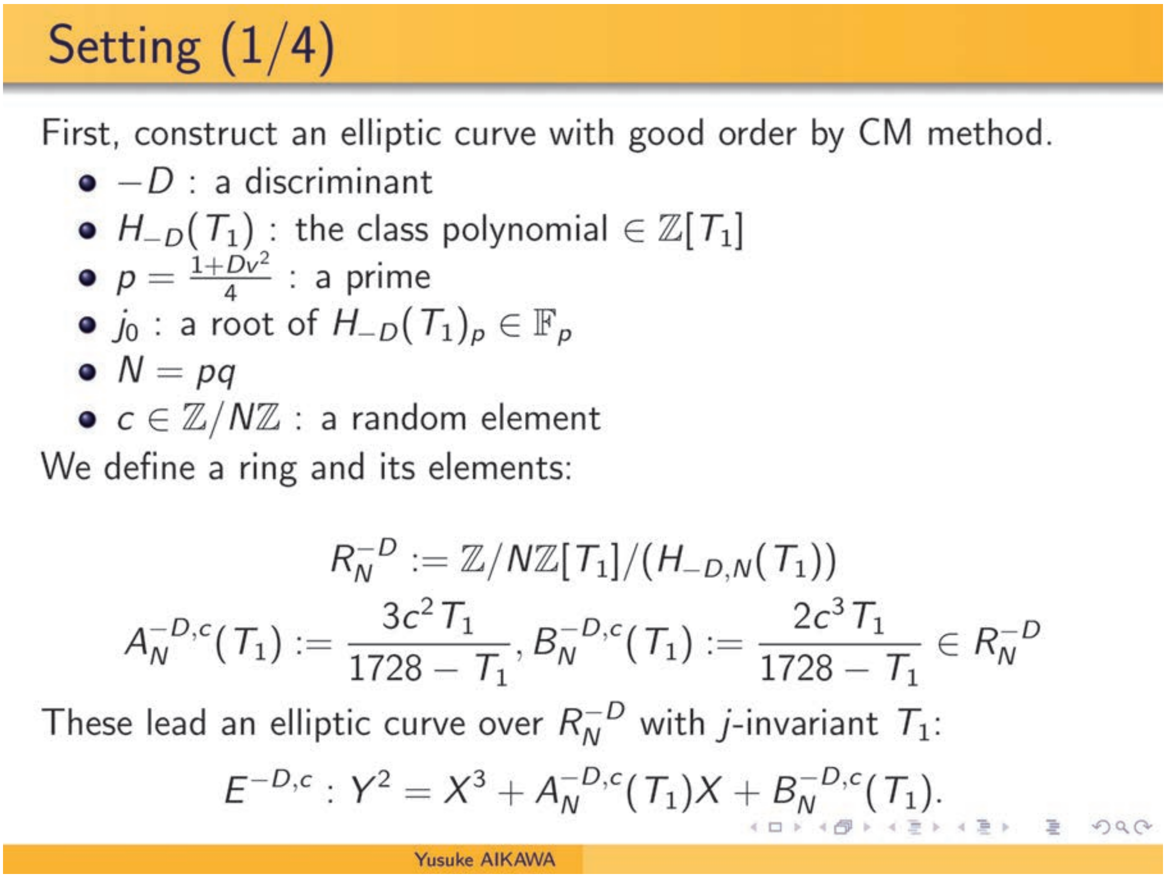


Outline of the Algorithm

$-D$: a discriminant with the class polynomial $H_{-D}(T)$
N : a composite having a prime factor $4p = 1 + v^2 D$
Attempt to factor N

1. Generate E with $\#E(\mathbb{F}_p) = p$ by using CM method
2. Generate a rational point P of E by extending the coefficient ring
3. Compute NP
4. Construct a system of reduction NP to integers

In SCIS2017, M. Shirase constructs for a discriminant with $\deg H_{-D}(T) \leq 2$, i.e.
$|D| \in \{3, 11, 19, 35, 43, 51, 67, 91, 115, 123, 163, 187, 235, 267, 403, 427\}$
$\rightsquigarrow$ In present study, delete the condition on the degree of $H_{-D}(T)$.



Setting (1/4)

First, construct an elliptic curve with good order by CM method.

- $-D$: a discriminant
- $H_{-D}(T_1)$: the class polynomial $\in \mathbb{Z}[T_1]$
- $p = \frac{1+Dv^2}{4}$: a prime
- j_0 : a root of $H_{-D}(T_1)_p \in \mathbb{F}_p$
- $N = pq$
- $c \in \mathbb{Z}/N\mathbb{Z}$: a random element

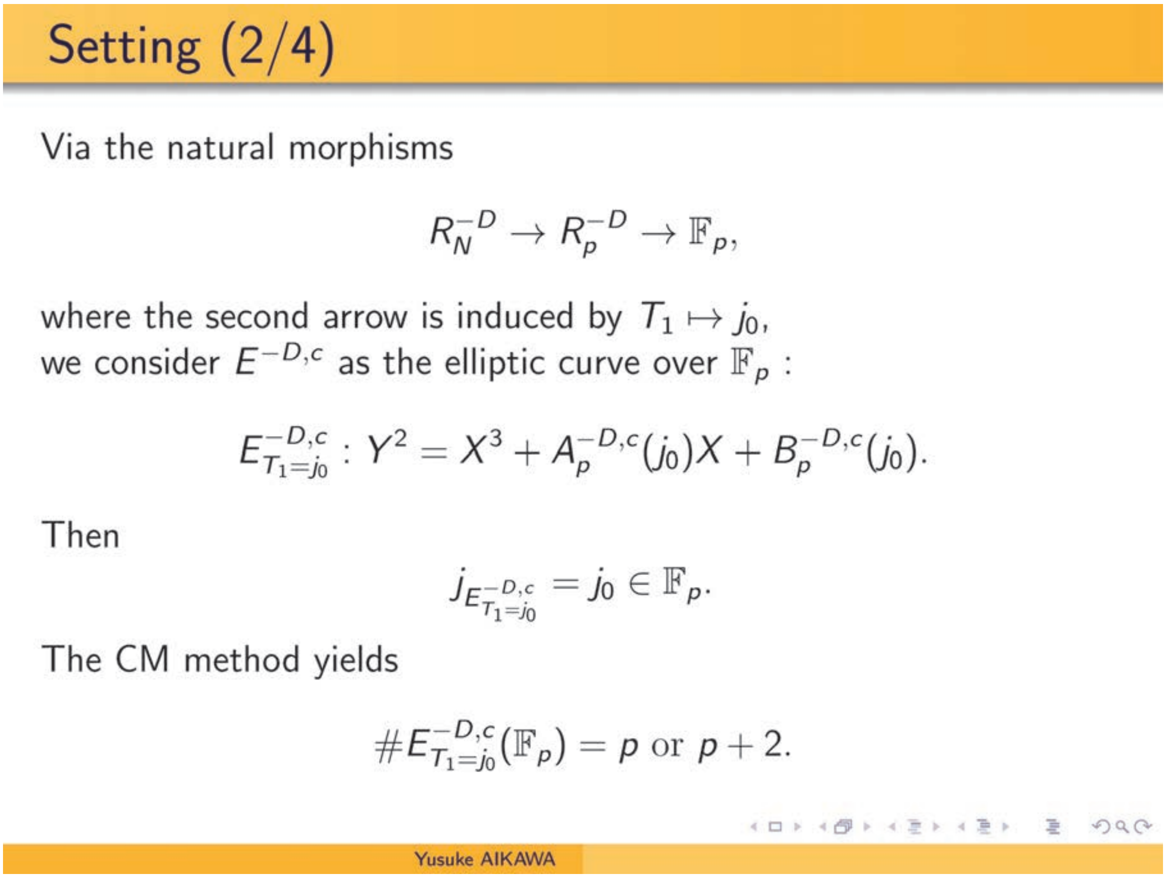
We define a ring and its elements:

$$R_N^{-D} := \mathbb{Z}/N\mathbb{Z}[T_1]/(H_{-D,N}(T_1))$$

$$A_N^{-D,c}(T_1) := \frac{3c^2T_1}{1728 - T_1}, B_N^{-D,c}(T_1) := \frac{2c^3T_1}{1728 - T_1} \in R_N^{-D}$$

These lead an elliptic curve over R_N^{-D} with j-invariant T_1:

$$E^{-D,c} : Y^2 = X^3 + A_N^{-D,c}(T_1)X + B_N^{-D,c}(T_1).$$



Setting (2/4)

Via the natural morphisms

$$R_N^{-D} \to R_p^{-D} \to \mathbb{F}_p,$$

where the second arrow is induced by $T_1 \mapsto j_0$,
we consider $E^{-D,c}$ as the elliptic curve over $\mathbb{F}_p$:

$$E_{T_1=j_0}^{-D,c} : Y^2 = X^3 + A_p^{-D,c}(j_0)X + B_p^{-D,c}(j_0).$$

Then

$$j_{E_{T_1=j_0}^{-D,c}} = j_0 \in \mathbb{F}_p.$$

The CM method yields

$$\#E_{T_1=j_0}^{-D,c}(\mathbb{F}_p) = p \text{ or } p + 2.$$

Setting (3/4)

Second, generate a rational point of $E^{-D,c}$ artificially.

- $x_0 \in \mathbb{Z}/N\mathbb{Z}$: a random element
- $\tau(T_1) := x_0^3 + A^{-D,c}(T_1)x_0 + B^{-D,c}(T_1) \in R_N^{-D}$

Set

$$\begin{aligned} S_N^{-D,\tau(T_1)} &:= R_N^{-D}[T_2]/(T_2^2 - \tau(T_1)) \\ &= \mathbb{Z}/N\mathbb{Z}[T_1, T_2]/(H_{-D,N}(T_1), T_2^2 - \tau(T_2)) \end{aligned}$$

Then

$$P := (x_0, T_2) \in E^{-D,c}(S_N^{-D,\tau(T_1)})$$

Setting (4/4)

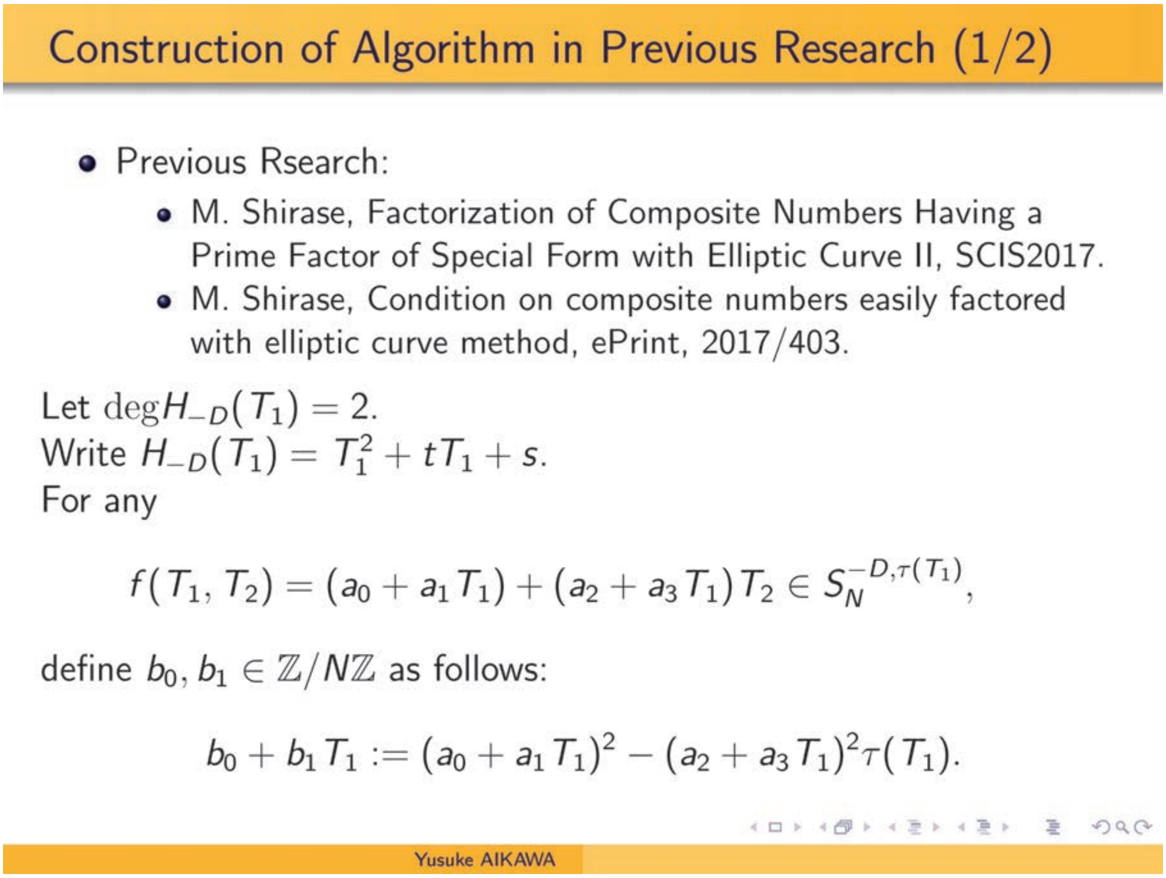
Take scalar multiplication by N:

$$NP = \left(\frac{\phi_N(x_0, T_2)}{\psi_N(x_0, T_2)^2}, \frac{\omega_N(x_0, T_2)}{\psi_N(x_0, T_2)^3}\right) \in E^{-D,c}(S_N^{-D,\tau(T_1)}).$$

Need to reduce the polynomial $d(T_1, T_2) := \psi_N(x_0, T_2)$ to an integer.
$\rightsquigarrow$ Suppose that $\tau(j_0) \in \mathbb{F}_p$ is a square, i.e. $\sigma^2 = \tau(j_0)$ for $\exists \sigma \in \mathbb{F}_p$,

$$\begin{aligned} &\Longrightarrow \quad S_N^{-D,\tau(T_1)} \to \mathbb{F}_p \ ; \ T_1 \mapsto j_0, \ T_2 \to \sigma, \\ &\Longrightarrow \quad E^{-D,c}(S_N^{-D,\tau(T_1)}) \to E_{T_1=j_0}^{-D,c}(\mathbb{F}_p) \ ; \ NP \mapsto \infty, \\ &\Longrightarrow \quad d(j_0, \sigma) = 0 \in \mathbb{F}_p. \end{aligned}$$

However, we need to compute the values j_0 and σ.



Construction of Algorithm in Previous Research (1/2)

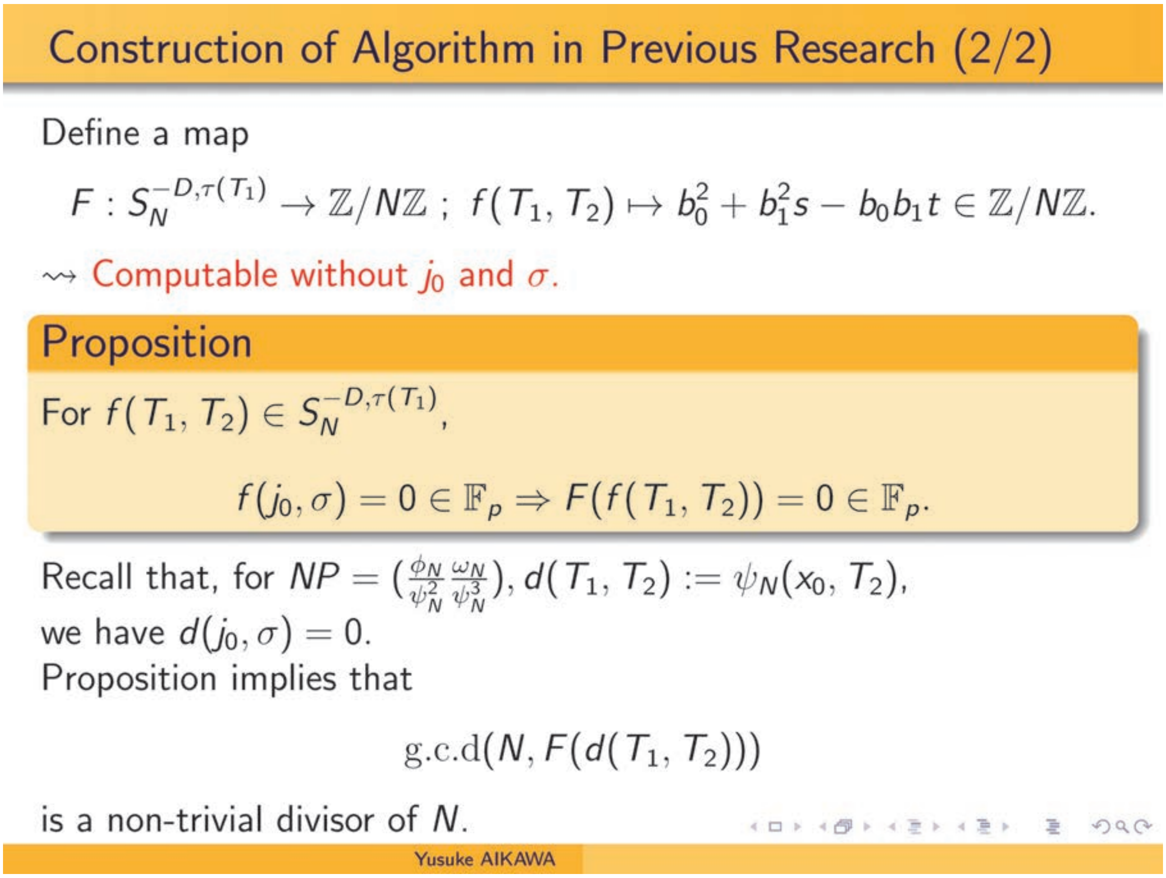
- Previous Rsearch:
  - M. Shirase, Factorization of Composite Numbers Having a Prime Factor of Special Form with Elliptic Curve II, SCIS2017.
  - M. Shirase, Condition on composite numbers easily factored with elliptic curve method, ePrint, 2017/403.

Let $\mathrm{deg} H_{-D}(T_1) = 2$.
Write $H_{-D}(T_1) = T_1^2 + tT_1 + s$.
For any

$$f(T_1, T_2) = (a_0 + a_1 T_1) + (a_2 + a_3 T_1) T_2 \in S_N^{-D,\tau(T_1)},$$

define $b_0, b_1 \in \mathbb{Z}/N\mathbb{Z}$ as follows:

$$b_0 + b_1 T_1 := (a_0 + a_1 T_1)^2 - (a_2 + a_3 T_1)^2 \tau(T_1).$$



Construction of Algorithm in Previous Research (2/2)

Define a map

$$F : S_N^{-D,\tau(T_1)} \to \mathbb{Z}/N\mathbb{Z} \ ; \ f(T_1, T_2) \mapsto b_0^2 + b_1^2 s - b_0 b_1 t \in \mathbb{Z}/N\mathbb{Z}.$$

$\rightsquigarrow$ Computable without j_0 and σ.

Proposition

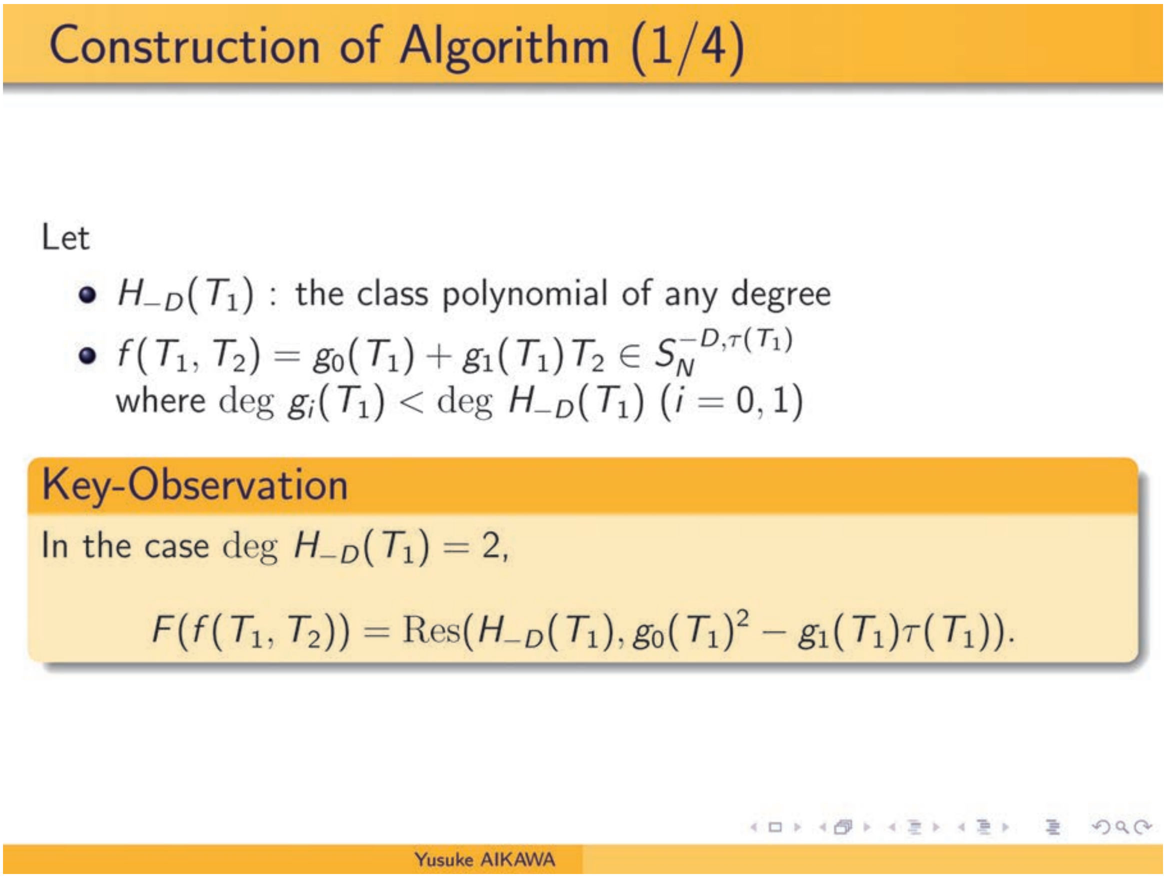
For $f(T_1, T_2) \in S_N^{-D,\tau(T_1)}$,

$$f(j_0, \sigma) = 0 \in \mathbb{F}_p \Rightarrow F(f(T_1, T_2)) = 0 \in \mathbb{F}_p.$$

Recall that, for $NP = (\frac{\phi_N}{\psi_N^2} \frac{\omega_N}{\psi_N^3})$, $d(T_1, T_2) := \psi_N(x_0, T_2)$,
we have $d(j_0, \sigma) = 0$.
Proposition implies that

$$\mathrm{g.c.d}(N, F(d(T_1, T_2)))$$

is a non-trivial divisor of N.



Construction of Algorithm (1/4)

Let

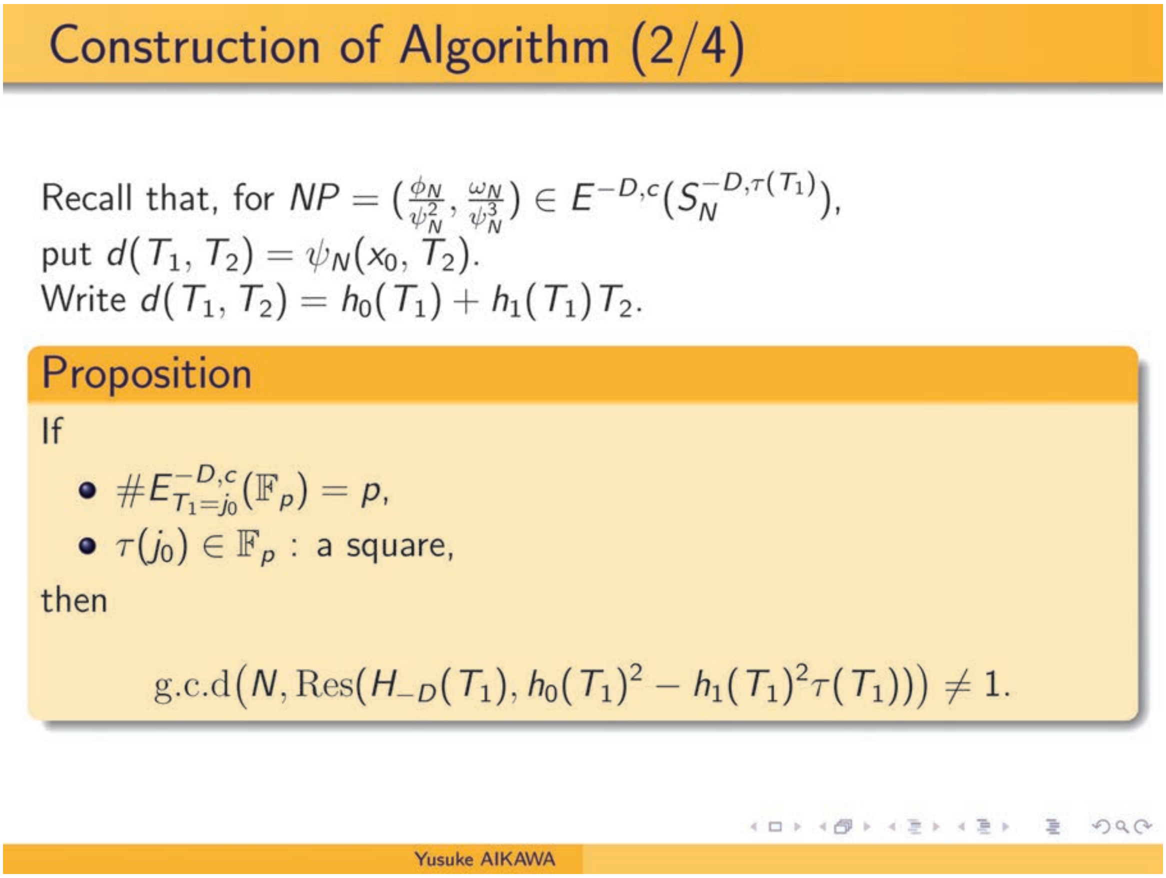
- $H_{-D}(T_1)$: the class polynomial of any degree
- $f(T_1, T_2) = g_0(T_1) + g_1(T_1)T_2 \in S_N^{-D,\tau(T_1)}$ where $\deg g_i(T_1) < \deg H_{-D}(T_1)$ $(i = 0, 1)$

Key-Observation

In the case $\deg H_{-D}(T_1) = 2$,

$$F(f(T_1, T_2)) = \mathrm{Res}(H_{-D}(T_1), g_0(T_1)^2 - g_1(T_1)\tau(T_1)).$$

Yusuke AIKAWA



Construction of Algorithm (2/4)

Recall that, for $NP = (\frac{\phi_N}{\psi_N^2}, \frac{\omega_N}{\psi_N^3}) \in E^{-D,c}(S_N^{-D,\tau(T_1)})$, put $d(T_1, T_2) = \psi_N(x_0, T_2)$. Write $d(T_1, T_2) = h_0(T_1) + h_1(T_1)T_2$.

Proposition

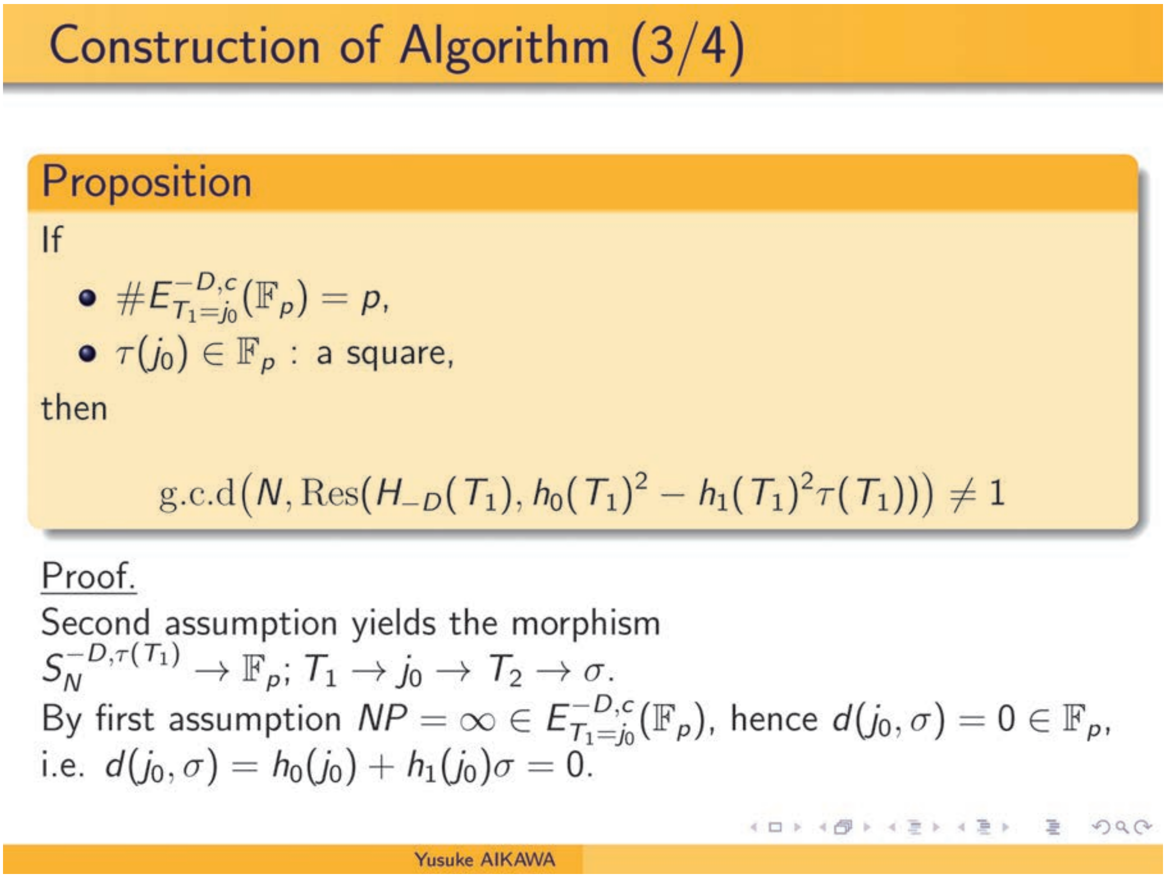
If

- $\# E^{-D,c}_{T_1=j_0}(\mathbb{F}_p) = p$,
- $\tau(j_0) \in \mathbb{F}_p$: a square,

then

$$\mathrm{g.c.d}(N, \mathrm{Res}(H_{-D}(T_1), h_0(T_1)^2 - h_1(T_1)^2\tau(T_1))) \neq 1.$$

Yusuke AIKAWA



Construction of Algorithm (3/4)

Proposition

If

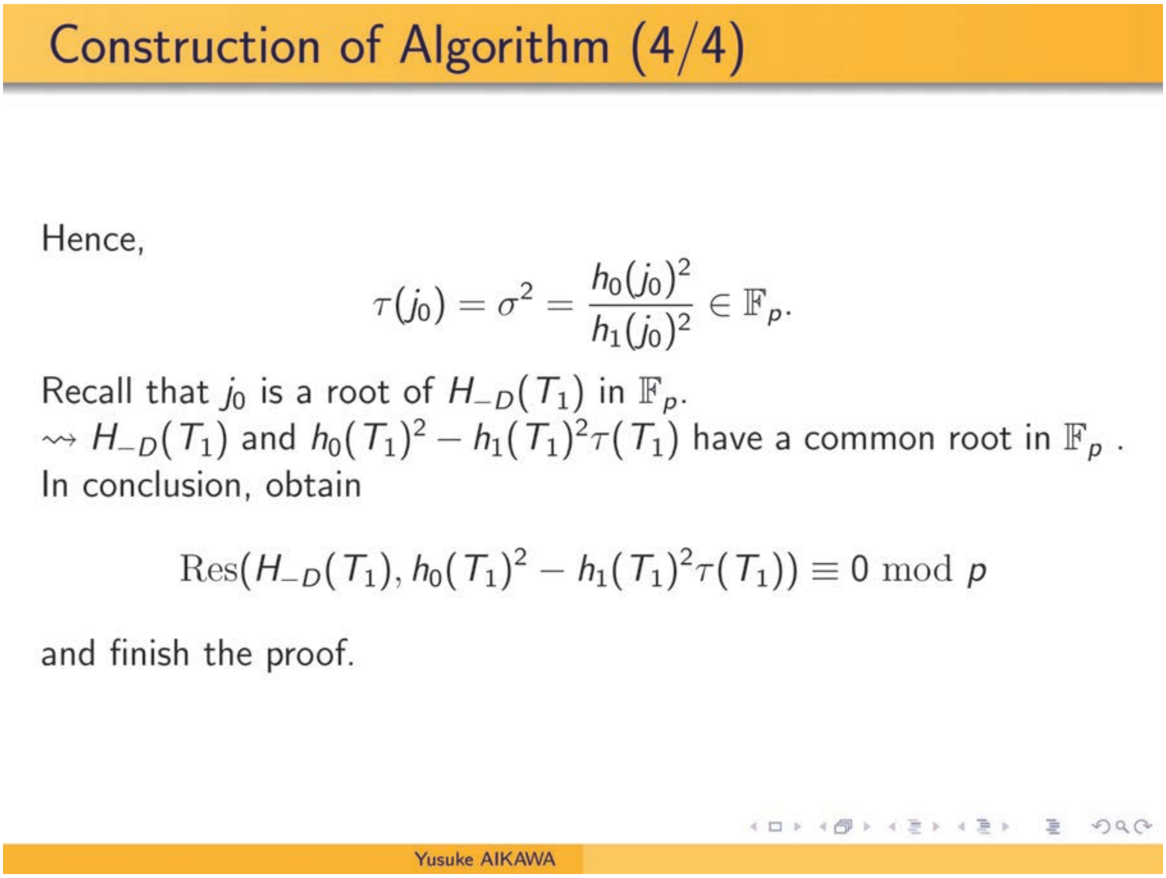
- $\#E^{-D,c}_{T_1=j_0}(\mathbb{F}_p) = p$,
- $\tau(j_0) \in \mathbb{F}_p$: a square,

then

$$\mathrm{g.c.d}\big(N, \mathrm{Res}(H_{-D}(T_1), h_0(T_1)^2 - h_1(T_1)^2\tau(T_1))\big) \neq 1$$

Proof.

Second assumption yields the morphism
$S_N^{-D,\tau(T_1)} \to \mathbb{F}_p;\ T_1 \to j_0 \to T_2 \to \sigma.$
By first assumption $NP = \infty \in E^{-D,c}_{T_1=j_0}(\mathbb{F}_p)$, hence $d(j_0, \sigma) = 0 \in \mathbb{F}_p$,
i.e. $d(j_0, \sigma) = h_0(j_0) + h_1(j_0)\sigma = 0$.



Construction of Algorithm (4/4)

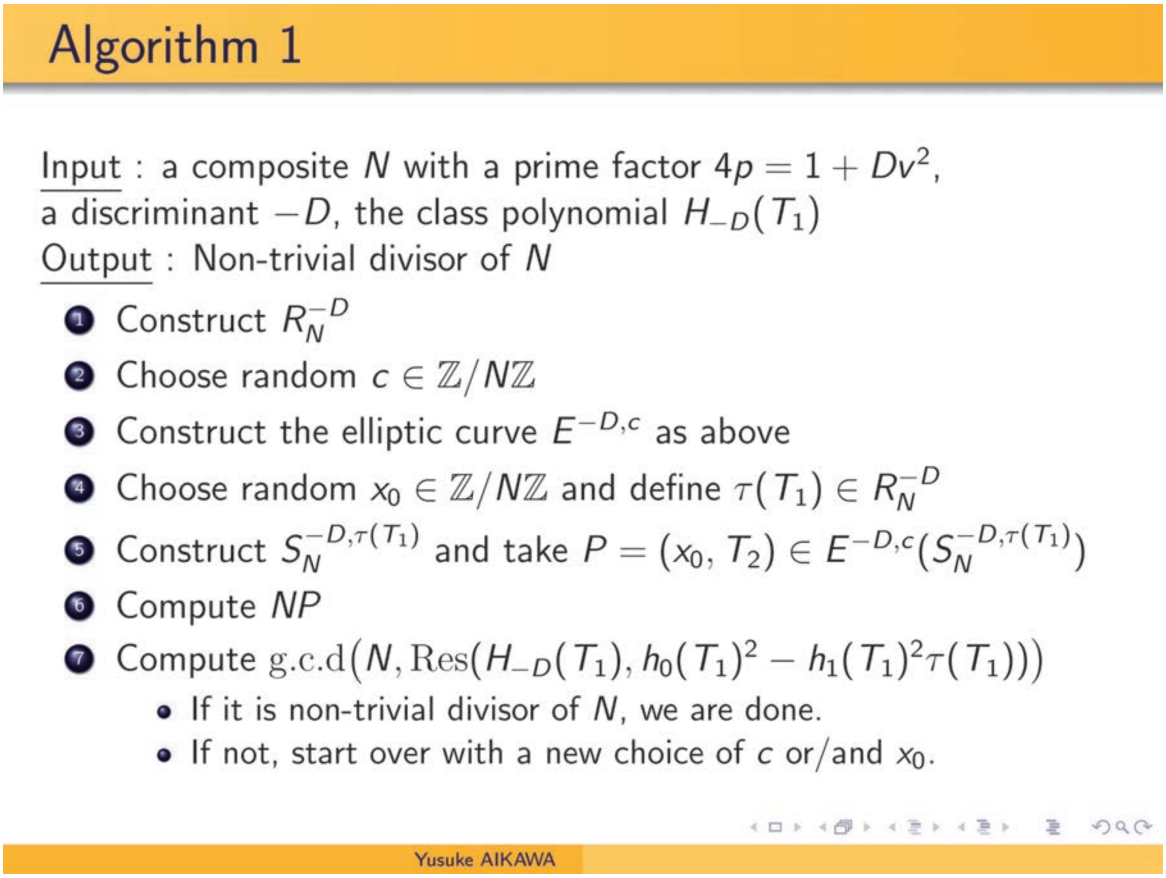
Hence,

$$\tau(j_0) = \sigma^2 = \frac{h_0(j_0)^2}{h_1(j_0)^2} \in \mathbb{F}_p.$$

Recall that j_0 is a root of $H_{-D}(T_1)$ in $\mathbb{F}_p$.
$\rightsquigarrow$ $H_{-D}(T_1)$ and $h_0(T_1)^2 - h_1(T_1)^2\tau(T_1)$ have a common root in $\mathbb{F}_p$.
In conclusion, obtain

$$\mathrm{Res}(H_{-D}(T_1), h_0(T_1)^2 - h_1(T_1)^2\tau(T_1)) \equiv 0 \bmod p$$

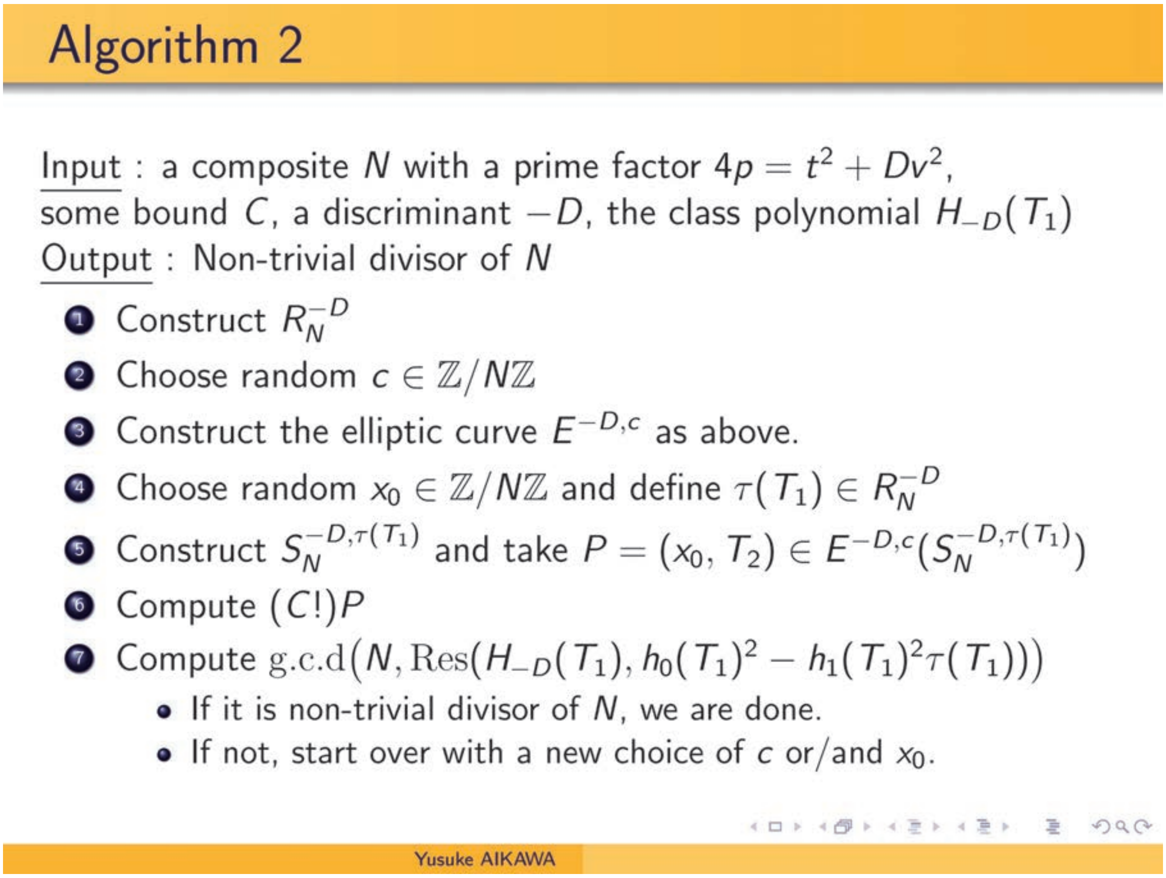
and finish the proof.



Algorithm 1

Input : a composite N with a prime factor $4p = 1 + Dv^2$, a discriminant $-D$, the class polynomial $H_{-D}(T_1)$
Output : Non-trivial divisor of N

1. Construct R_N^{-D}
2. Choose random $c \in \mathbb{Z}/N\mathbb{Z}$
3. Construct the elliptic curve $E^{-D,c}$ as above
4. Choose random $x_0 \in \mathbb{Z}/N\mathbb{Z}$ and define $\tau(T_1) \in R_N^{-D}$
5. Construct $S_N^{-D,\tau(T_1)}$ and take $P = (x_0, T_2) \in E^{-D,c}(S_N^{-D,\tau(T_1)})$
6. Compute NP
7. Compute $\mathrm{g.c.d}\big(N, \mathrm{Res}(H_{-D}(T_1), h_0(T_1)^2 - h_1(T_1)^2\tau(T_1))\big)$
   - If it is non-trivial divisor of N, we are done.
   - If not, start over with a new choice of c or/and x_0.



Algorithm 2

Input : a composite N with a prime factor $4p = t^2 + Dv^2$, some bound C, a discriminant $-D$, the class polynomial $H_{-D}(T_1)$
Output : Non-trivial divisor of N

1. Construct R_N^{-D}
2. Choose random $c \in \mathbb{Z}/N\mathbb{Z}$
3. Construct the elliptic curve $E^{-D,c}$ as above.
4. Choose random $x_0 \in \mathbb{Z}/N\mathbb{Z}$ and define $\tau(T_1) \in R_N^{-D}$
5. Construct $S_N^{-D,\tau(T_1)}$ and take $P = (x_0, T_2) \in E^{-D,c}(S_N^{-D,\tau(T_1)})$
6. Compute $(C!)P$
7. Compute $\mathrm{g.c.d}\big(N, \mathrm{Res}(H_{-D}(T_1), h_0(T_1)^2 - h_1(T_1)^2\tau(T_1))\big)$
   - If it is non-trivial divisor of N, we are done.
   - If not, start over with a new choice of c or/and x_0.

Numerical Examples

- $-D = -23, \deg H_{-D}(T) = 3$
  $4p = 4 \times 570942088504121 = 1210134^2 + D \times 9961456^2$
  $p + 1 - t = 570942087293988 \mid 2000!$
  $q = 883478470161233$
  $N = p \times q = 504415042902280115530654941193$
- $-D = -56, \deg H_{-D}(T) = 4$
  $4p = 4 \times 804161 = 450^2 + D \times 232^2$
  $p + 1 - 450 = 2^5 \times 3 \times 7 \times 13 \times 23$
  $N = p \times q = 488391904291$
- $-D = -131, \deg H_{-D}(T) = 5$
  $4p = 1 + D \times 139116657084339^2$ (case $t = 1$)
  $q = 868610670601296908562434196197$
  $N = p \times q = 550547418976985666816226779885030$
  $828558826986967578267955611$

Future Works

- Experiment
- Theoretical estimate

A modification of the discrete Fourier transform for the code defined by Garcia-Stichtenoth tower

Norihiro Nakashima
(Tokyo Denki University)

work with Hajime Matsui

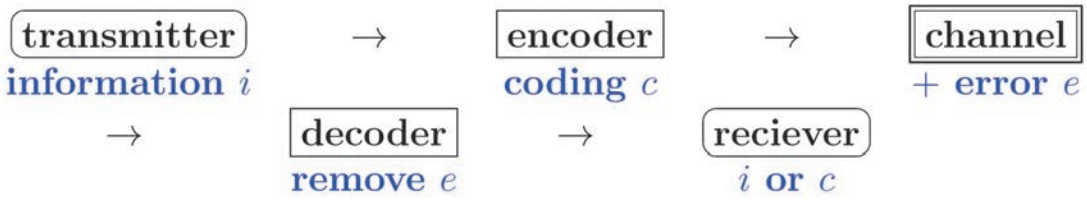
Workshop on analysis of mathematical cryptography via algebraic methods

2018.2.6

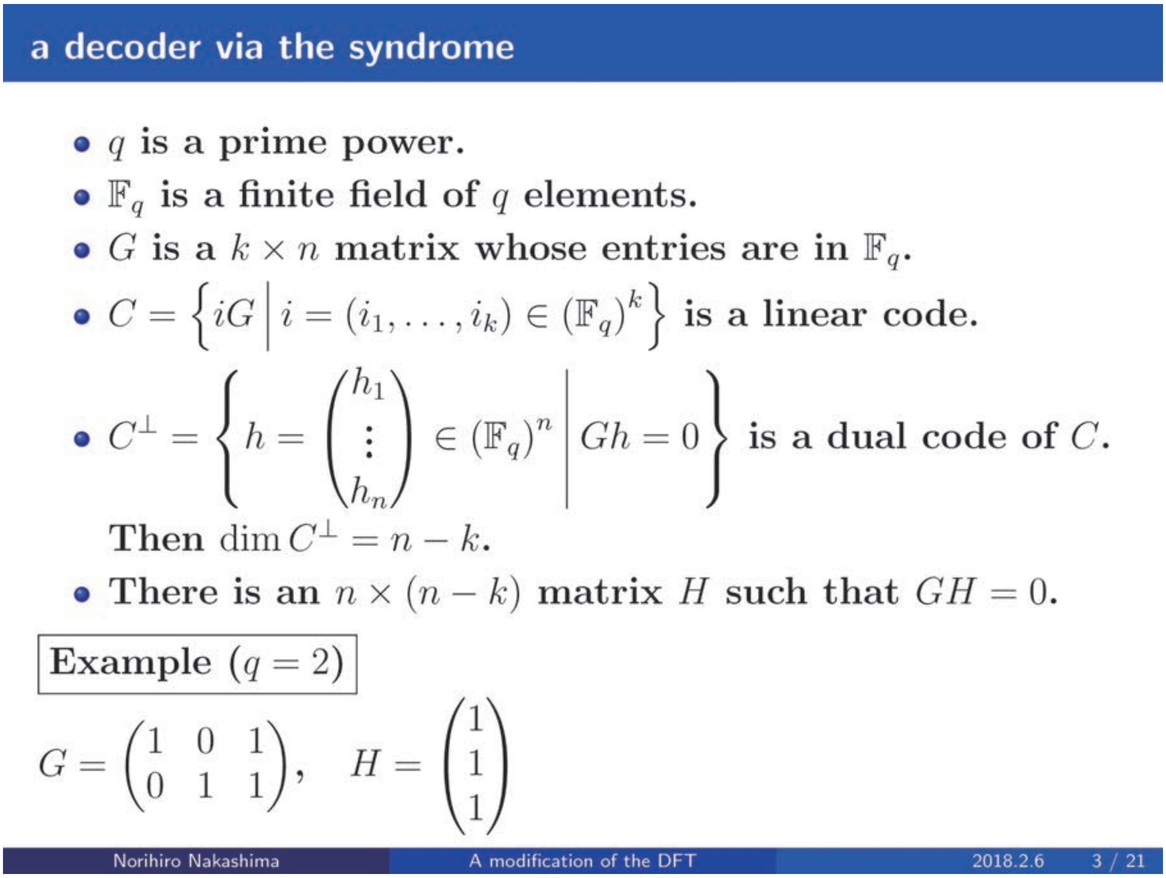
Error correcting codes

Aim

Recover the original information i.



- In an encoder, the original information is changed to a recoverable codeword.
- Errors occur in a noisy channel, and the codeword is changed to $r = c + e$.
- Errors are removed in a decoder.



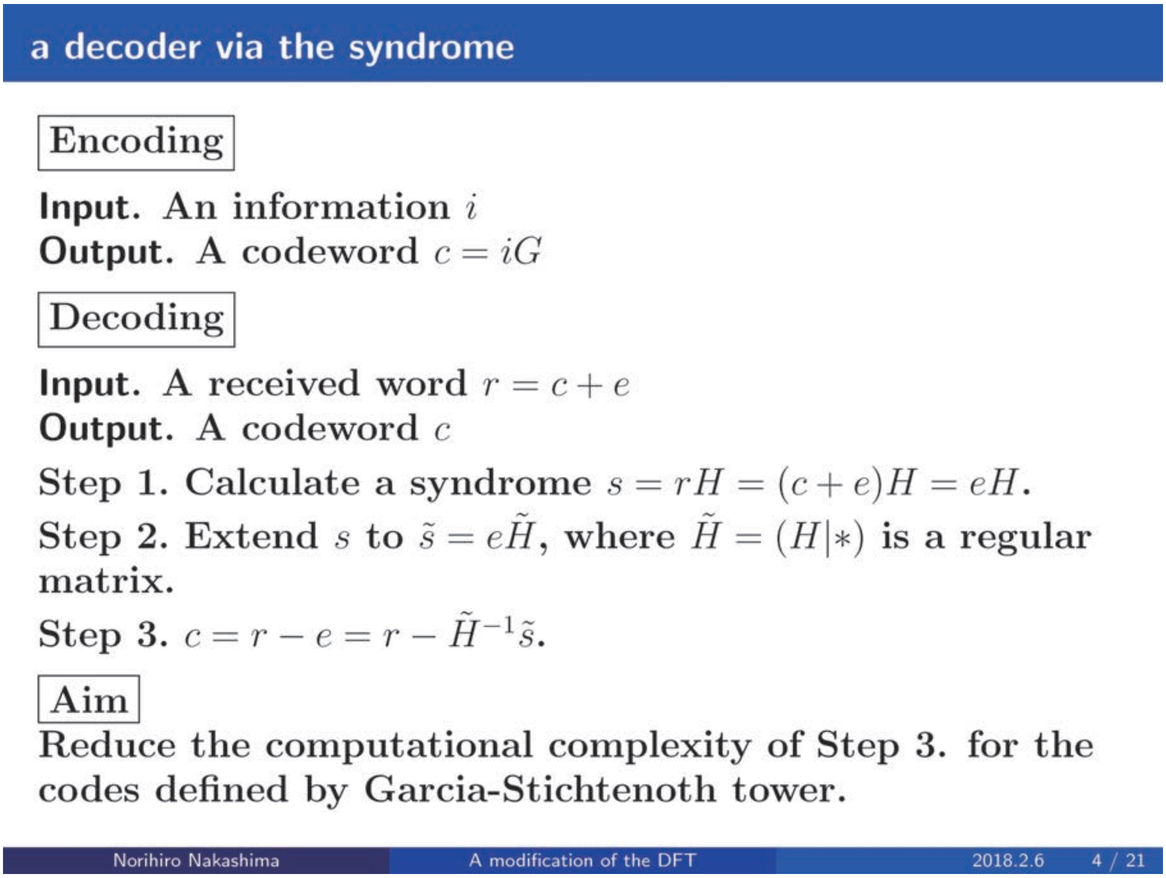
a decoder via the syndrome

- q is a prime power.
- $\mathbb{F}_q$ is a finite field of q elements.
- G is a $k \times n$ matrix whose entries are in $\mathbb{F}_q$.
- $C = \left\{ iG \,\middle|\, i = (i_1, \ldots, i_k) \in (\mathbb{F}_q)^k \right\}$ is a linear code.
- $C^\perp = \left\{ h = \begin{pmatrix} h_1 \\ \vdots \\ h_n \end{pmatrix} \in (\mathbb{F}_q)^n \,\middle|\, Gh = 0 \right\}$ is a dual code of C.

  Then $\dim C^\perp = n - k$.
- There is an $n \times (n-k)$ matrix H such that $GH = 0$.

Example $(q = 2)$

$$G = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}, \quad H = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$



a decoder via the syndrome

Encoding

Input. An information i
Output. A codeword $c = iG$

Decoding

Input. A received word $r = c + e$
Output. A codeword c

Step 1. Calculate a syndrome $s = rH = (c + e)H = eH$.

Step 2. Extend s to $\tilde{s} = e\tilde{H}$, where $\tilde{H} = (H|*)$ is a regular matrix.

Step 3. $c = r - e = r - \tilde{H}^{-1}\tilde{s}$.

Aim

Reduce the computational complexity of Step 3. for the codes defined by Garcia-Stichtenoth tower.

Discrete Fourier transform

- $(\mathbb{F}_q)^\Omega = \left\{ (v_\omega)_{\omega\in\Omega} \,\middle|\, v_\omega \in \mathbb{F}_q \right\}$ **for a finite set** Ω**.**
- $[i] = \{0, 1, \ldots, i\}$ **for a positive integer** i**.**
- **Denote** $0^0 = 1$**.**

Definition

For $\Psi \subseteq \mathbb{F}_q^m$ *and* $B \subseteq [q-1]^m$*, we define*

$$\mathcal{F}_{\Psi,B} : \mathbb{F}_q^\Psi \to \mathbb{F}_q^B,\ \left(c_{\underline{\psi}}\right)_{\underline{\psi}\in\Psi} \mapsto \left(\sum_{\underline{\psi}\in\Psi} c_{\underline{\psi}} \underline{\psi}^{\underline{b}} \right)_{\underline{b}\in B}.$$

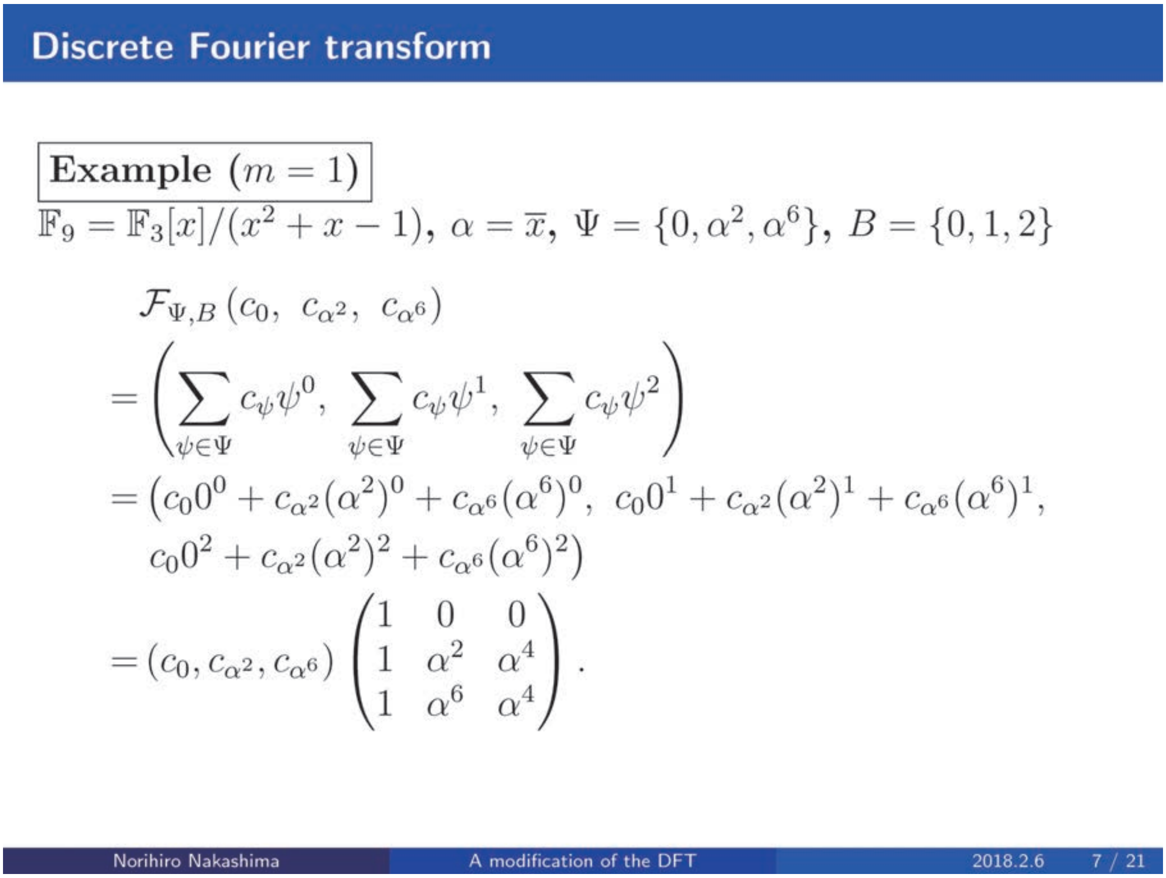
We call $\mathcal{F}_{\Psi,B}$ *a discrete Fourier transform (DFT).*

Discrete Fourier transform

$\boxed{\textbf{Example } (m=1)}$

$\Psi = \{\psi_1, \psi_2, \psi_3\} \subseteq \mathbb{F}_9$, $B = \{0, 1, 2\} \subseteq [9-1]$

$$\begin{aligned}
&\mathcal{F}_{\Psi,B}\left(c_{\psi_1},\ c_{\psi_2},\ c_{\psi_3}\right) \\
&= \left(\sum_{\psi\in\Psi} c_\psi \psi^0,\ \sum_{\psi\in\Psi} c_\psi \psi^1,\ \sum_{\psi\in\Psi} c_\psi \psi^2 \right) \\
&= \left(c_{\psi_1}\psi_1^0 + c_{\psi_2}\psi_2^0 + c_{\psi_3}\psi_3^0,\ c_{\psi_1}\psi_1^1 + c_{\psi_2}\psi_2^1 + c_{\psi_3}\psi_3^1,\right. \\
&\quad \left. c_{\psi_1}\psi_1^2 + c_{\psi_2}\psi_1^2 + c_{\psi_3}\psi_3^2 \right) \\
&= \left(c_{\psi_1}, c_{\psi_2}, c_{\psi_3}\right) \begin{pmatrix} 1 & \psi_1 & \psi_1^2 \\ 1 & \psi_2 & \psi_2^2 \\ 1 & \psi_3 & \psi_3^2 \end{pmatrix}.
\end{aligned}$$

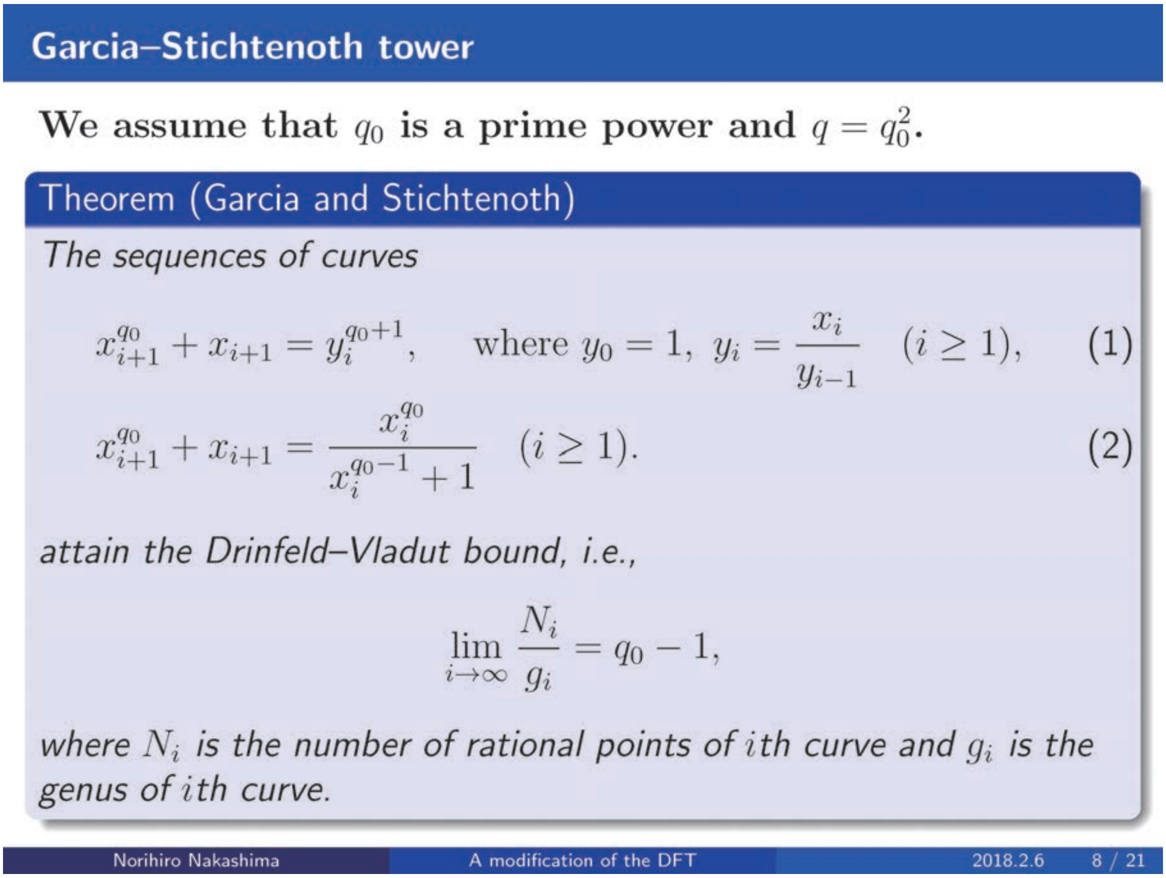


Discrete Fourier transform

$\boxed{\textbf{Example } (m=1)}$

$\mathbb{F}_9 = \mathbb{F}_3[x]/(x^2+x-1)$, $\alpha = \overline{x}$, $\Psi = \{0, \alpha^2, \alpha^6\}$, $B = \{0,1,2\}$

$$
\begin{aligned}
&\mathcal{F}_{\Psi,B}(c_0,\ c_{\alpha^2},\ c_{\alpha^6}) \\
&= \left(\sum_{\psi\in\Psi} c_\psi \psi^0,\ \sum_{\psi\in\Psi} c_\psi \psi^1,\ \sum_{\psi\in\Psi} c_\psi \psi^2 \right) \\
&= (c_0 0^0 + c_{\alpha^2}(\alpha^2)^0 + c_{\alpha^6}(\alpha^6)^0,\ c_0 0^1 + c_{\alpha^2}(\alpha^2)^1 + c_{\alpha^6}(\alpha^6)^1, \\
&\quad c_0 0^2 + c_{\alpha^2}(\alpha^2)^2 + c_{\alpha^6}(\alpha^6)^2) \\
&= (c_0, c_{\alpha^2}, c_{\alpha^6}) \begin{pmatrix} 1 & 0 & 0 \\ 1 & \alpha^2 & \alpha^4 \\ 1 & \alpha^6 & \alpha^4 \end{pmatrix}.
\end{aligned}
$$



Garcia–Stichtenoth tower

We assume that q_0 is a prime power and $q = q_0^2$.

Theorem (Garcia and Stichtenoth)

The sequences of curves

$$x_{i+1}^{q_0} + x_{i+1} = y_i^{q_0+1}, \qquad \text{where } y_0 = 1,\ y_i = \frac{x_i}{y_{i-1}} \quad (i \geq 1), \tag{1}$$

$$x_{i+1}^{q_0} + x_{i+1} = \frac{x_i^{q_0}}{x_i^{q_0-1}+1} \quad (i \geq 1). \tag{2}$$

attain the Drinfeld–Vladut bound, i.e.,

$$\lim_{i\to\infty} \frac{N_i}{g_i} = q_0 - 1,$$

where N_i is the number of rational points of ith curve and g_i is the genus of ith curve.

Codes defined by the Garcia–Stichtenoth tower

In the rest of this talk, let
$\Psi = \left\{ \underline{\psi} \in \left(\mathbb{F}_q^\times\right)^m \middle| \psi_{i+1}^{q_0} + \psi_{i+1} = y_i^{q_0+1}(\underline{\psi}) \text{ for } 1 \le i \le m-1 \right\}$**.**

Definition

For a positive integer l*,*
$B = \{\underline{b} = (b_1, \ldots, b_m) \in [q-2]^m \mid \sum_{i=1}^m o(x_i) b_i \le l\}$*, where* $o(x_i)$ *is the pole order of* x_i*. Then*

$$C(B, \Psi) = \left\{ \left(c_{\underline{\psi}}\right)_{\underline{\psi}\in\Psi} \in \mathbb{F}_q^\Psi \,\middle|\, \mathcal{F}_{\Psi,B}\left(\left(c_{\underline{\psi}}\right)_{\underline{\psi}\in\Psi}\right) = \mathbf{0} \right\}$$

is an algebraic geometric code defined by the Garcia–Stichtenoth tower.

Codes defined by the Garcia–Stichtenoth tower

Fact

It is known that a decoding algorithm via the Berlekamp–Massey–Sakata algorithm and the discrete Fourier transform can be applied to $C(B, \Psi)$**.**
If the number t **of error entries satisfies**

$$t \le \frac{N_i - k - g_i + 1}{2} = g_i \frac{(N_i/g_i) - (k/g_i) - 1 + (1/g_i)}{2},$$

then the decoding algorithm computes error word correctly, where k **is the dimension of the dual code of** $C(B, \Psi)$**.**

Calculation of DFTs

- $V_1 = \{\psi \in \mathbb{F}_q \mid \psi^{q_0} + \psi = 1\}$

Then $|V_1| = q_0$.

$\boxed{\textbf{Example}}$

$\mathbb{F}_9 = \mathbb{F}_3[x]/(x^2 + x - 1)$, $\alpha = \overline{x}$.
$V_1 = \{\alpha^4, \alpha^5, \alpha^7\}$.

Lemma

$$\Psi = \left\{ \left(\psi_j \prod_{k=1}^{j-1} \psi_k^{q_0+1}\right)_{j=1}^{m} \middle| \psi_1 \in (\mathbb{F}_q)^\times, \psi_2, \ldots, \psi_m \in V_1 \right\}.$$

Calculation of DFTs

$$\boxed{x_{i+1}^{q_0} + x_{i+1} = y_i^{q_0+1}, y_0 = 1, \ y_i = x_i / y_{i-1}}$$

$\boxed{\textbf{Example } (m = 3, q_0 = 3)}$

$\underline{\psi} = (\psi_1, \psi_2\psi_1^4, \psi_3\psi_2^4\psi_1^4) \in \Psi$ **for** $\psi_1 \in \mathbb{F}_9^\times$, $\psi_2, \psi_3 \in V_1$. **Indeed**

$$\begin{aligned}
&(\psi_2\psi_1^4)^3 + \psi_2\psi_1^4 \\
=&\psi_2^3\psi_1^4 + \psi_2\psi_1^4 = (\psi_2^3 + \psi_2)\psi_1^4 = \psi_1^4 \\
=&(\psi_1/y_0)^4 = y_1(\underline{\psi})^4,
\end{aligned}$$

$$\begin{aligned}
&(\psi_3\psi_2^4\psi_1^4)^3 + \psi_3\psi_2^4\psi_1^4 \\
=&\psi_3^3\psi_2^4\psi_1^4 + \psi_3\psi_2^4\psi_1^4 = (\psi_3^3 + \psi_3)\psi_2^4\psi_1^4 = \psi_2^4\psi_1^4 \\
=&\psi_2^4\psi_1^{12} = \psi_2^4\psi_1^{16}/\psi_1^4 = (x_2(\underline{\psi})/y_1(\underline{\psi}))^4.
\end{aligned}$$

Calculation of DFTs

We may consider that Ψ is $\mathbb{F}_q^\times \times V_1^{m-1}$ as index sets.

Example $(m=2)$

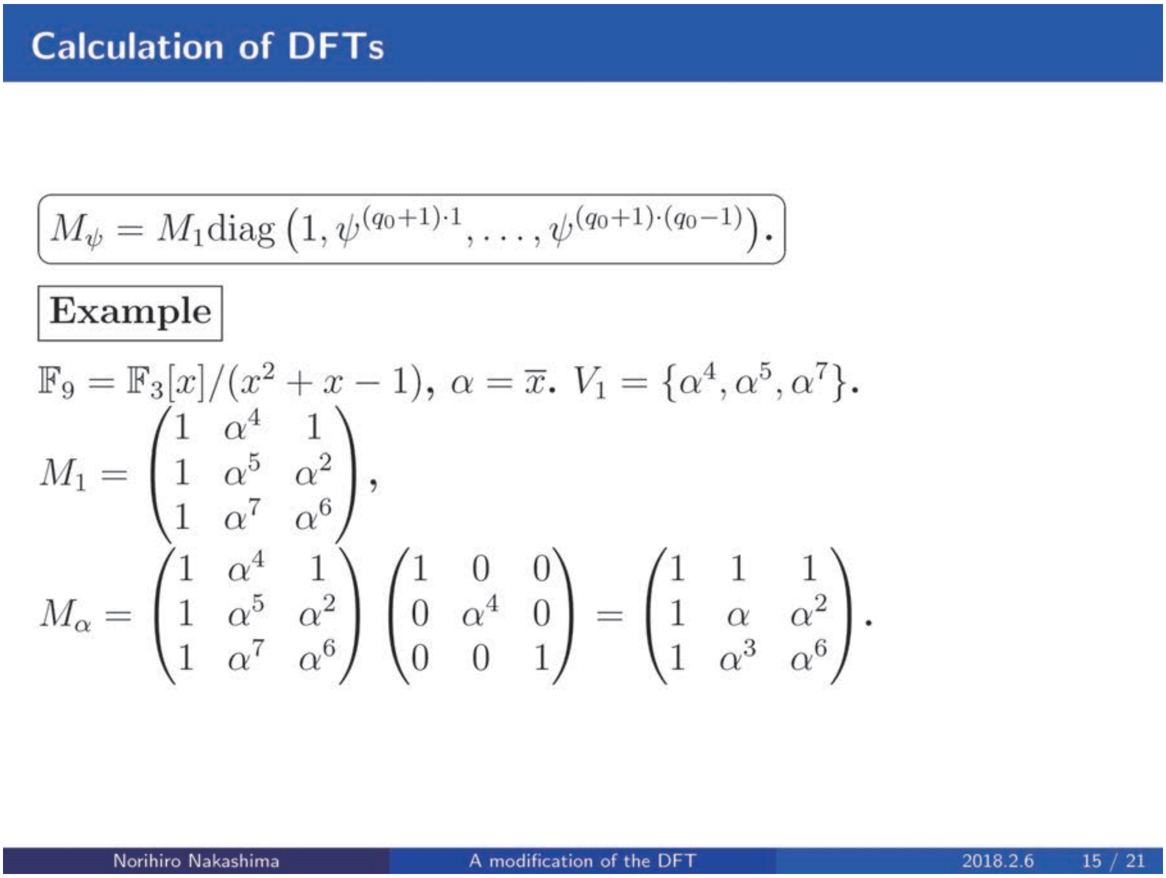
$\mathbb{F}_9 = \mathbb{F}_3[x]/(x^2+x-1)$, $\alpha = \overline{x}$.

	1	α	α^2	α^3	α^4	α^5	α^6	α^7
1					●	●		●
α	●	●		●				
α^2					●	●		●
α^3	●	●		●				
α^4					●	●		●
α^5	●	●		●				
α^6					●	●		●
α^7	●	●		●				

- **Points are elements in Ψ.**
- $|\Psi| = q_0^{m+1} = 27$.

Calculation of DFTs

- $M_1 = \begin{pmatrix} 1 & v_1 & \cdots & v_1^{q_0-1} \\ 1 & v_2 & \cdots & v_2^{q_0-1} \\ \vdots & \vdots & & \vdots \\ 1 & v_{q_0} & \cdots & v_{q_0}^{q_0-1} \end{pmatrix}$, **where** $V_1 = \{v_1, v_2, \ldots, v_{q_0}\}$.
- **For** $\psi \in \mathbb{F}_q^\times \setminus \{1\}$, **we define** $M_\psi = M_1 \mathrm{diag}\left(1, \psi^{(q_0+1)\cdot 1}, \ldots, \psi^{(q_0+1)\cdot(q_0-1)}\right)$.
- **The inverse matrix of M_ψ is obtained by** $M_\psi^{-1} = \mathrm{diag}\left(1, \psi^{(-q_0-1)\cdot 1}, \ldots, \psi^{(-q_0-1)\cdot(q_0-1)}\right) M_1^{-1}$.



Calculation of DFTs

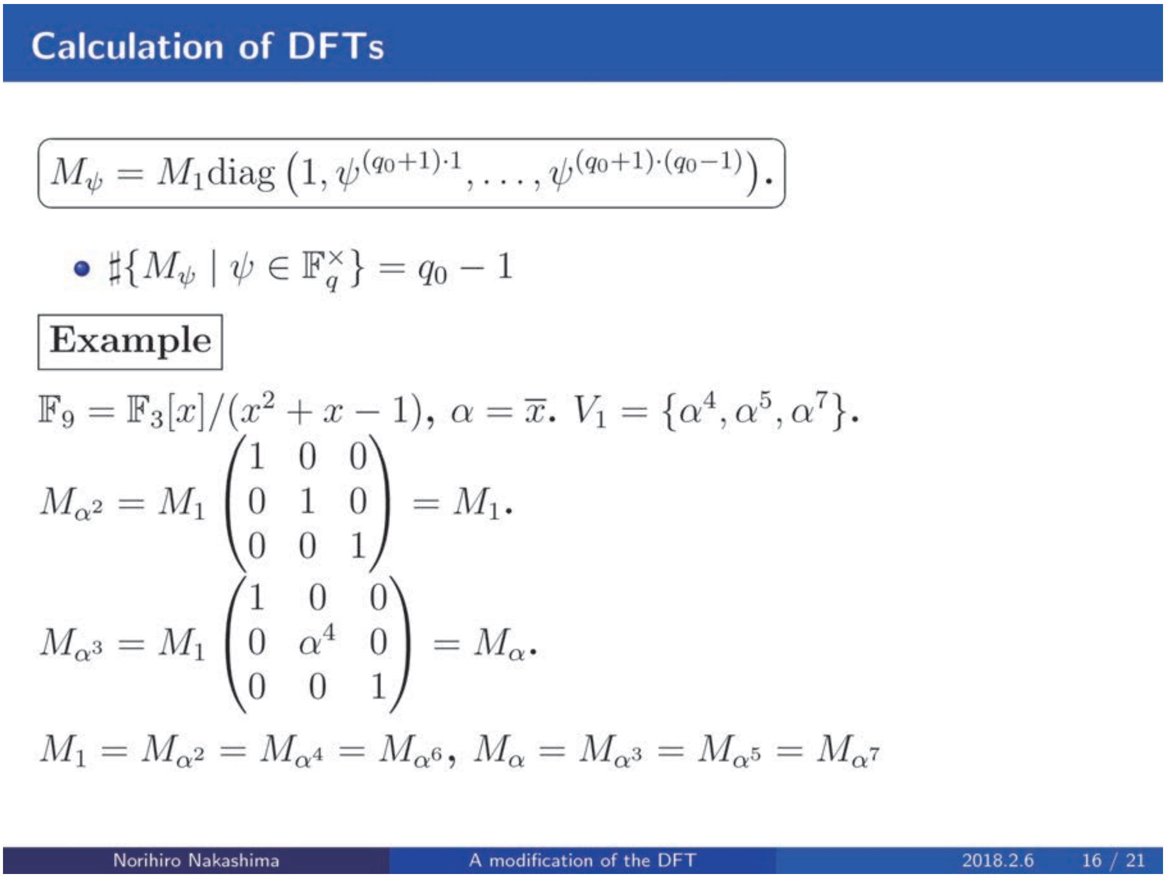
$$\boxed{M_\psi = M_1 \mathrm{diag}\left(1, \psi^{(q_0+1)\cdot 1}, \ldots, \psi^{(q_0+1)\cdot(q_0-1)}\right).}$$

Example

$\mathbb{F}_9 = \mathbb{F}_3[x]/(x^2+x-1)$, $\alpha = \overline{x}$. $V_1 = \{\alpha^4, \alpha^5, \alpha^7\}$.

$$M_1 = \begin{pmatrix} 1 & \alpha^4 & 1 \\ 1 & \alpha^5 & \alpha^2 \\ 1 & \alpha^7 & \alpha^6 \end{pmatrix},$$

$$M_\alpha = \begin{pmatrix} 1 & \alpha^4 & 1 \\ 1 & \alpha^5 & \alpha^2 \\ 1 & \alpha^7 & \alpha^6 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \alpha^4 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & \alpha & \alpha^2 \\ 1 & \alpha^3 & \alpha^6 \end{pmatrix}.$$



Calculation of DFTs

$$\boxed{M_\psi = M_1 \mathrm{diag}\left(1, \psi^{(q_0+1)\cdot 1}, \ldots, \psi^{(q_0+1)\cdot(q_0-1)}\right).}$$

- $\sharp\{M_\psi \mid \psi \in \mathbb{F}_q^\times\} = q_0 - 1$

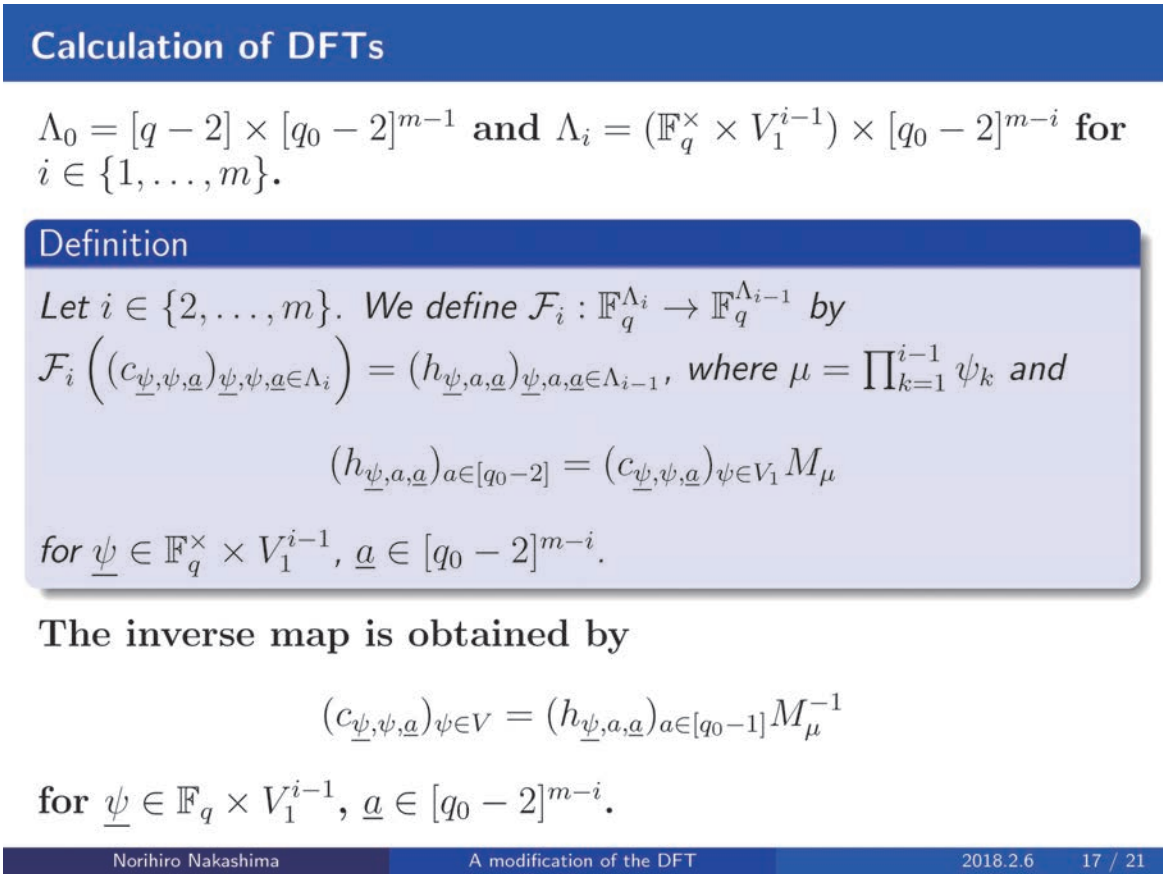
Example

$\mathbb{F}_9 = \mathbb{F}_3[x]/(x^2+x-1)$, $\alpha = \overline{x}$. $V_1 = \{\alpha^4, \alpha^5, \alpha^7\}$.

$$M_{\alpha^2} = M_1 \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = M_1.$$

$$M_{\alpha^3} = M_1 \begin{pmatrix} 1 & 0 & 0 \\ 0 & \alpha^4 & 0 \\ 0 & 0 & 1 \end{pmatrix} = M_\alpha.$$

$$M_1 = M_{\alpha^2} = M_{\alpha^4} = M_{\alpha^6},\ M_\alpha = M_{\alpha^3} = M_{\alpha^5} = M_{\alpha^7}$$



Calculation of DFTs

$\Lambda_0 = [q-2] \times [q_0-2]^{m-1}$ **and** $\Lambda_i = (\mathbb{F}_q^\times \times V_1^{i-1}) \times [q_0-2]^{m-i}$ **for** $i \in \{1, \ldots, m\}$.

Definition

Let $i \in \{2, \ldots, m\}$. *We define* $\mathcal{F}_i : \mathbb{F}_q^{\Lambda_i} \to \mathbb{F}_q^{\Lambda_{i-1}}$ *by* $\mathcal{F}_i\left((c_{\underline{\psi},\psi,\underline{a}})_{\underline{\psi},\psi,\underline{a}\in\Lambda_i}\right) = (h_{\underline{\psi},a,\underline{a}})_{\underline{\psi},a,\underline{a}\in\Lambda_{i-1}}$, *where* $\mu = \prod_{k=1}^{i-1} \psi_k$ *and*

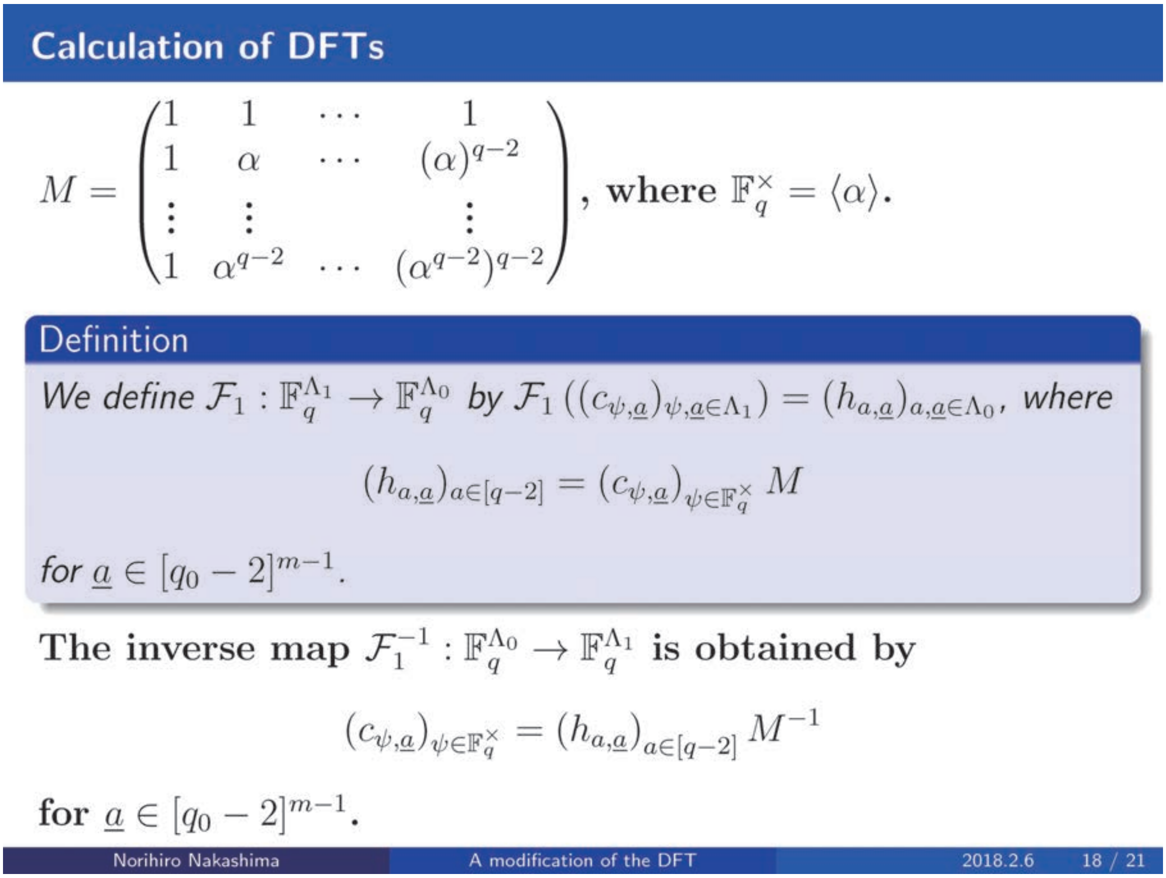
$$(h_{\underline{\psi},a,\underline{a}})_{a\in[q_0-2]} = (c_{\underline{\psi},\psi,\underline{a}})_{\psi\in V_1} M_\mu$$

for $\underline{\psi} \in \mathbb{F}_q^\times \times V_1^{i-1}$, $\underline{a} \in [q_0-2]^{m-i}$.

The inverse map is obtained by

$$(c_{\underline{\psi},\psi,\underline{a}})_{\psi\in V} = (h_{\underline{\psi},a,\underline{a}})_{a\in[q_0-1]} M_\mu^{-1}$$

for $\underline{\psi} \in \mathbb{F}_q \times V_1^{i-1}$, $\underline{a} \in [q_0-2]^{m-i}$.



Calculation of DFTs

$$M = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & \alpha & \cdots & (\alpha)^{q-2} \\ \vdots & \vdots & & \vdots \\ 1 & \alpha^{q-2} & \cdots & (\alpha^{q-2})^{q-2} \end{pmatrix}, \text{ where } \mathbb{F}_q^\times = \langle\alpha\rangle.$$

Definition

We define $\mathcal{F}_1 : \mathbb{F}_q^{\Lambda_1} \to \mathbb{F}_q^{\Lambda_0}$ *by* $\mathcal{F}_1\left((c_{\psi,\underline{a}})_{\psi,\underline{a}\in\Lambda_1}\right) = (h_{a,\underline{a}})_{a,\underline{a}\in\Lambda_0}$, *where*

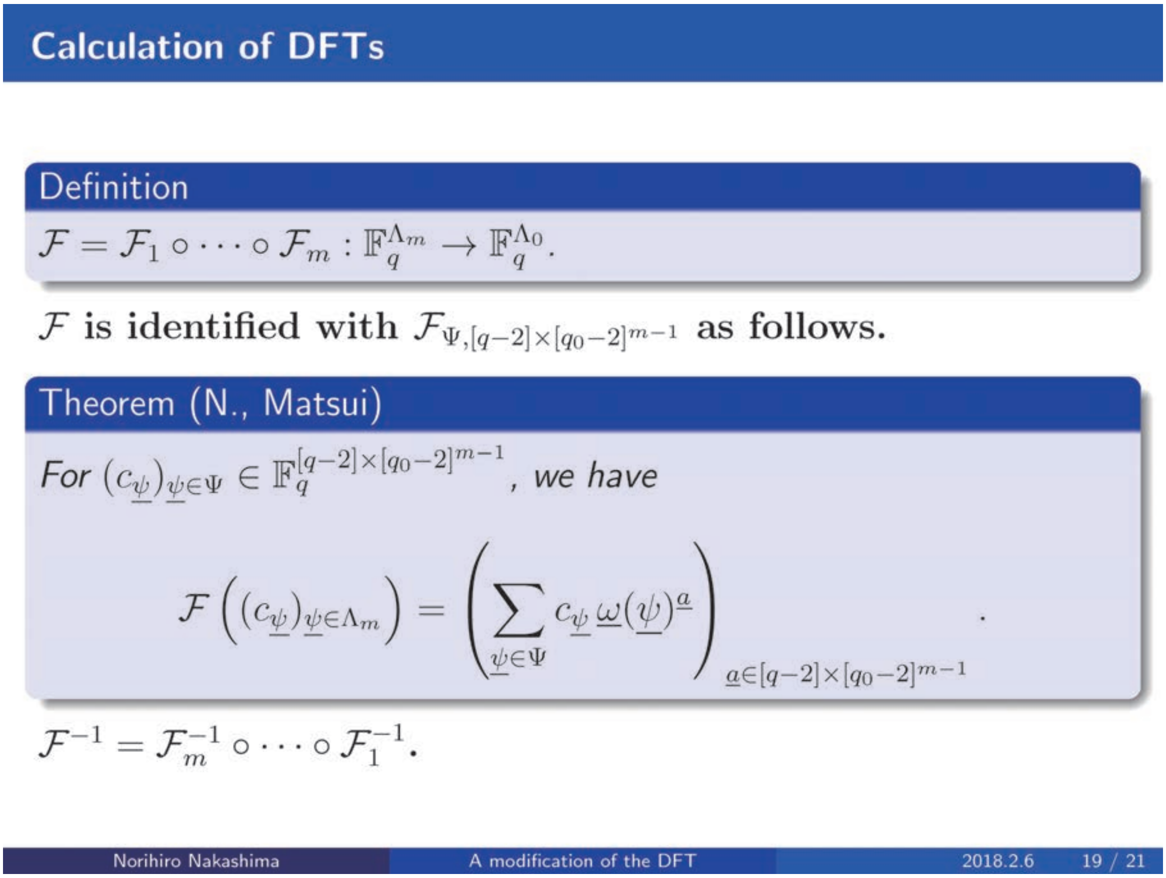
$$(h_{a,\underline{a}})_{a\in[q-2]} = (c_{\psi,\underline{a}})_{\psi\in\mathbb{F}_q^\times} M$$

for $\underline{a} \in [q_0-2]^{m-1}$.

The inverse map $\mathcal{F}_1^{-1} : \mathbb{F}_q^{\Lambda_0} \to \mathbb{F}_q^{\Lambda_1}$ **is obtained by**

$$(c_{\psi,\underline{a}})_{\psi\in\mathbb{F}_q^\times} = (h_{a,\underline{a}})_{a\in[q-2]} M^{-1}$$

for $\underline{a} \in [q_0-2]^{m-1}$.



Calculation of DFTs

Definition

$\mathcal{F} = \mathcal{F}_1 \circ \cdots \circ \mathcal{F}_m : \mathbb{F}_q^{\Lambda_m} \to \mathbb{F}_q^{\Lambda_0}$.

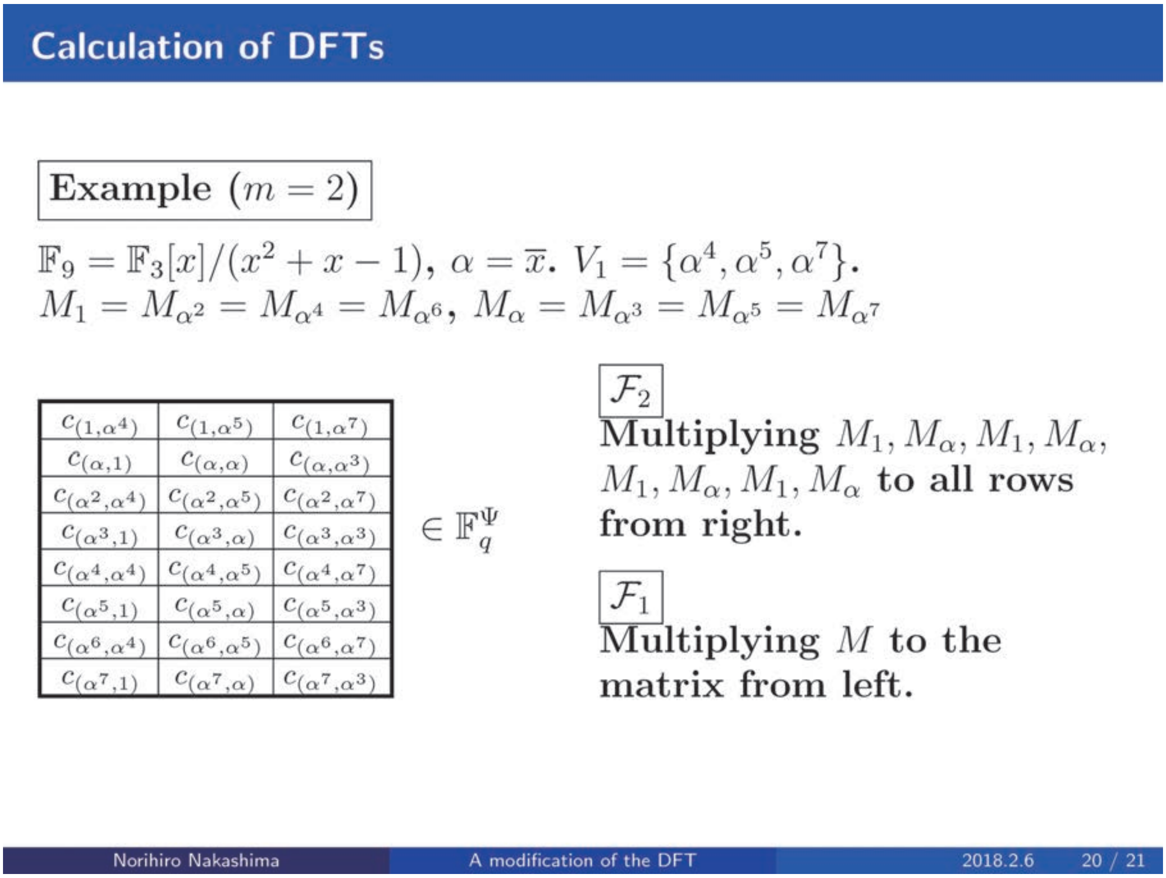
$\mathcal{F}$ **is identified with** $\mathcal{F}_{\Psi,[q-2]\times[q_0-2]^{m-1}}$ **as follows.**

Theorem (N., Matsui)

For $(c_{\underline{\psi}})_{\underline{\psi}\in\Psi} \in \mathbb{F}_q^{[q-2]\times[q_0-2]^{m-1}}$*, we have*

$$\mathcal{F}\left((c_{\underline{\psi}})_{\underline{\psi}\in\Lambda_m}\right) = \left(\sum_{\underline{\psi}\in\Psi} c_{\underline{\psi}}\, \underline{\omega}(\underline{\psi})^{\underline{a}}\right)_{\underline{a}\in[q-2]\times[q_0-2]^{m-1}}.$$

$\mathcal{F}^{-1} = \mathcal{F}_m^{-1} \circ \cdots \circ \mathcal{F}_1^{-1}$.



Calculation of DFTs

$\boxed{\text{Example } (m = 2)}$

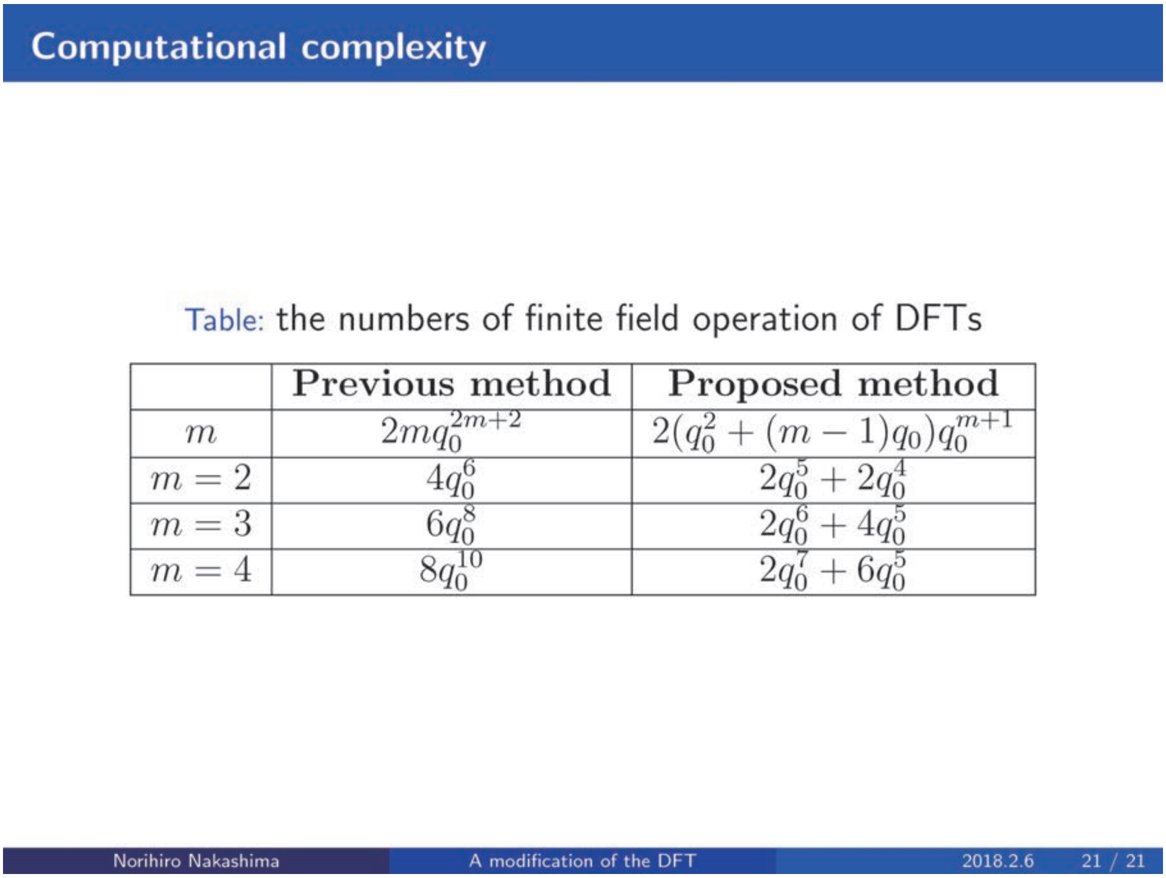
$\mathbb{F}_9 = \mathbb{F}_3[x]/(x^2 + x - 1)$, $\alpha = \overline{x}$. $V_1 = \{\alpha^4, \alpha^5, \alpha^7\}$.
$M_1 = M_{\alpha^2} = M_{\alpha^4} = M_{\alpha^6}$, $M_\alpha = M_{\alpha^3} = M_{\alpha^5} = M_{\alpha^7}$

$c_{(1,\alpha^4)}$	$c_{(1,\alpha^5)}$	$c_{(1,\alpha^7)}$
$c_{(\alpha,1)}$	$c_{(\alpha,\alpha)}$	$c_{(\alpha,\alpha^3)}$
$c_{(\alpha^2,\alpha^4)}$	$c_{(\alpha^2,\alpha^5)}$	$c_{(\alpha^2,\alpha^7)}$
$c_{(\alpha^3,1)}$	$c_{(\alpha^3,\alpha)}$	$c_{(\alpha^3,\alpha^3)}$
$c_{(\alpha^4,\alpha^4)}$	$c_{(\alpha^4,\alpha^5)}$	$c_{(\alpha^4,\alpha^7)}$
$c_{(\alpha^5,1)}$	$c_{(\alpha^5,\alpha)}$	$c_{(\alpha^5,\alpha^3)}$
$c_{(\alpha^6,\alpha^4)}$	$c_{(\alpha^6,\alpha^5)}$	$c_{(\alpha^6,\alpha^7)}$
$c_{(\alpha^7,1)}$	$c_{(\alpha^7,\alpha)}$	$c_{(\alpha^7,\alpha^3)}$

$\in \mathbb{F}_q^{\Psi}$

$\boxed{\mathcal{F}_2}$
Multiplying $M_1, M_\alpha, M_1, M_\alpha, M_1, M_\alpha, M_1, M_\alpha$ **to all rows from right.**

$\boxed{\mathcal{F}_1}$
Multiplying M **to the matrix from left.**



Computational complexity

Table: the numbers of finite field operation of DFTs

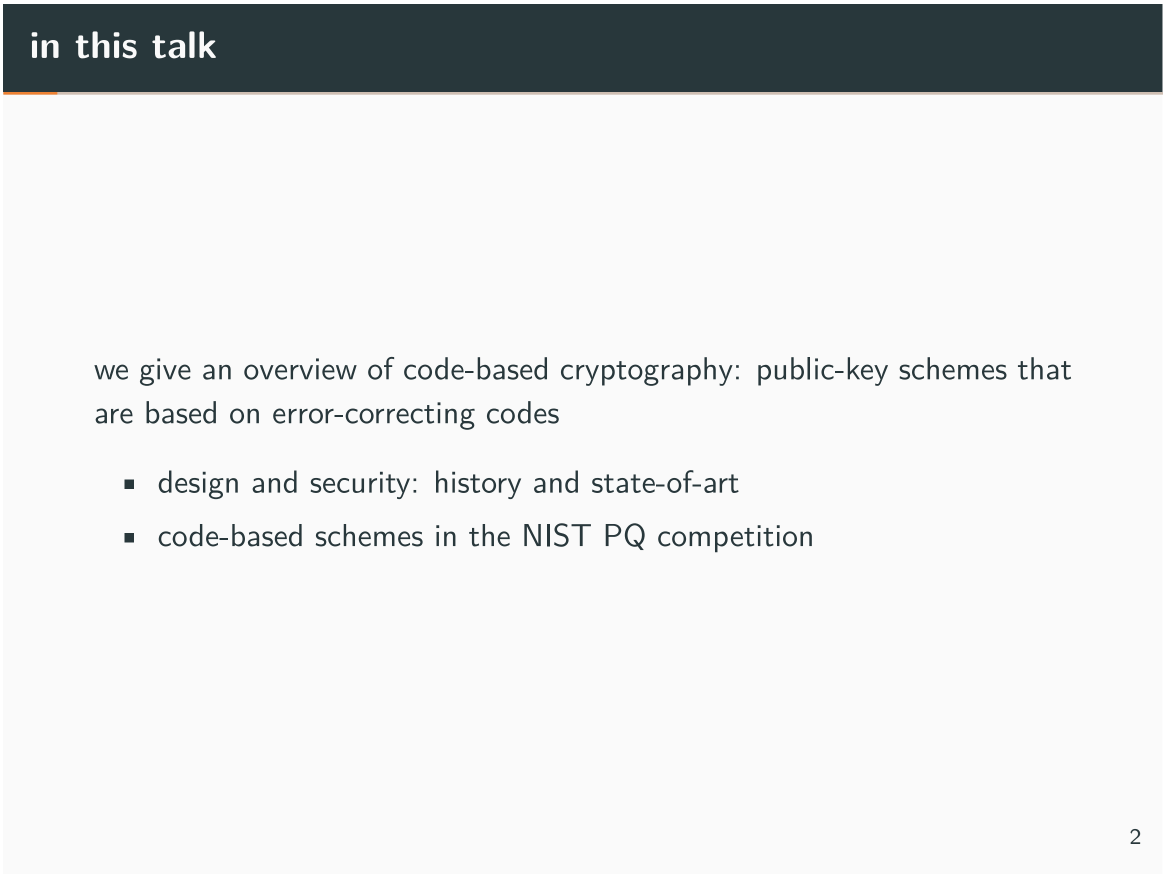
	Previous method	**Proposed method**
m	$2mq_0^{2m+2}$	$2(q_0^2+(m-1)q_0)q_0^{m+1}$
$m=2$	$4q_0^6$	$2q_0^5+2q_0^4$
$m=3$	$6q_0^8$	$2q_0^6+4q_0^5$
$m=4$	$8q_0^{10}$	$2q_0^7+6q_0^5$

Code-based cryptography: design and security

Carlos Cid

6 Feb 2018

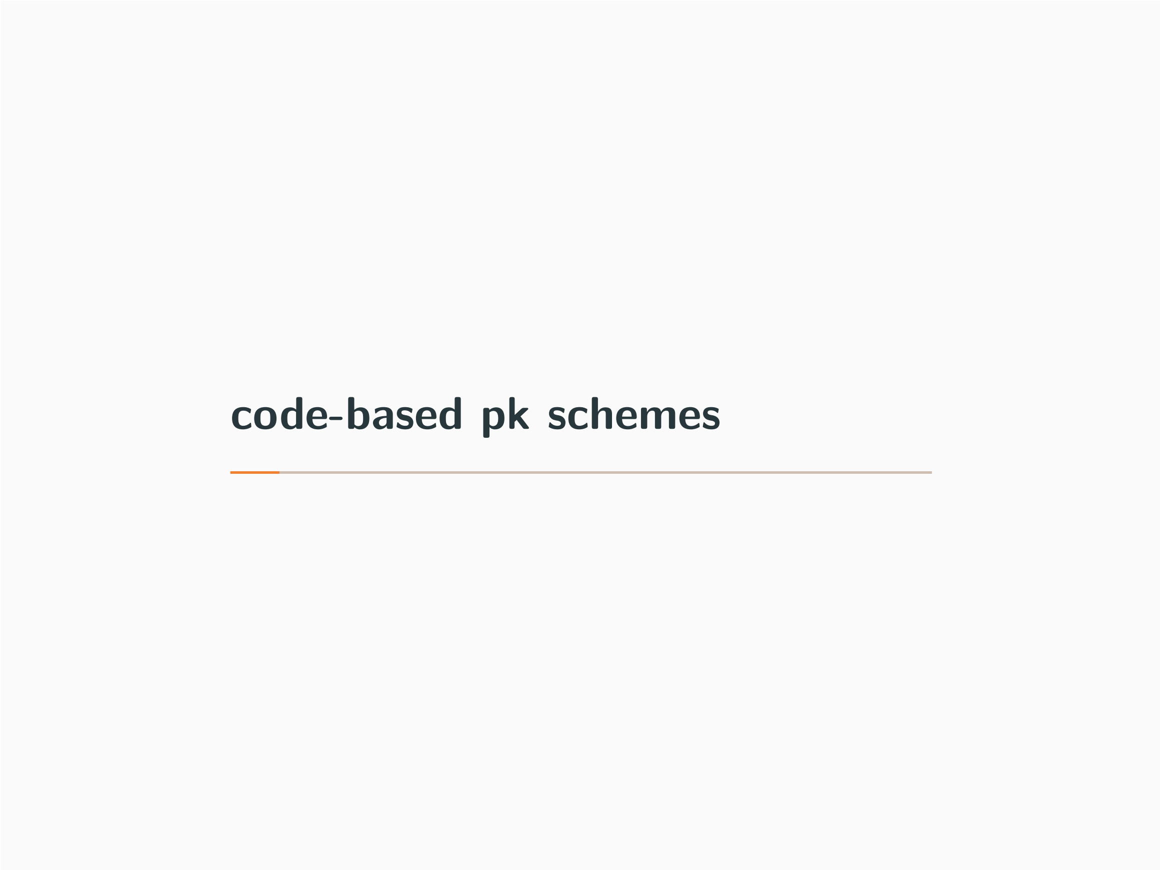
Royal Holloway, University of London



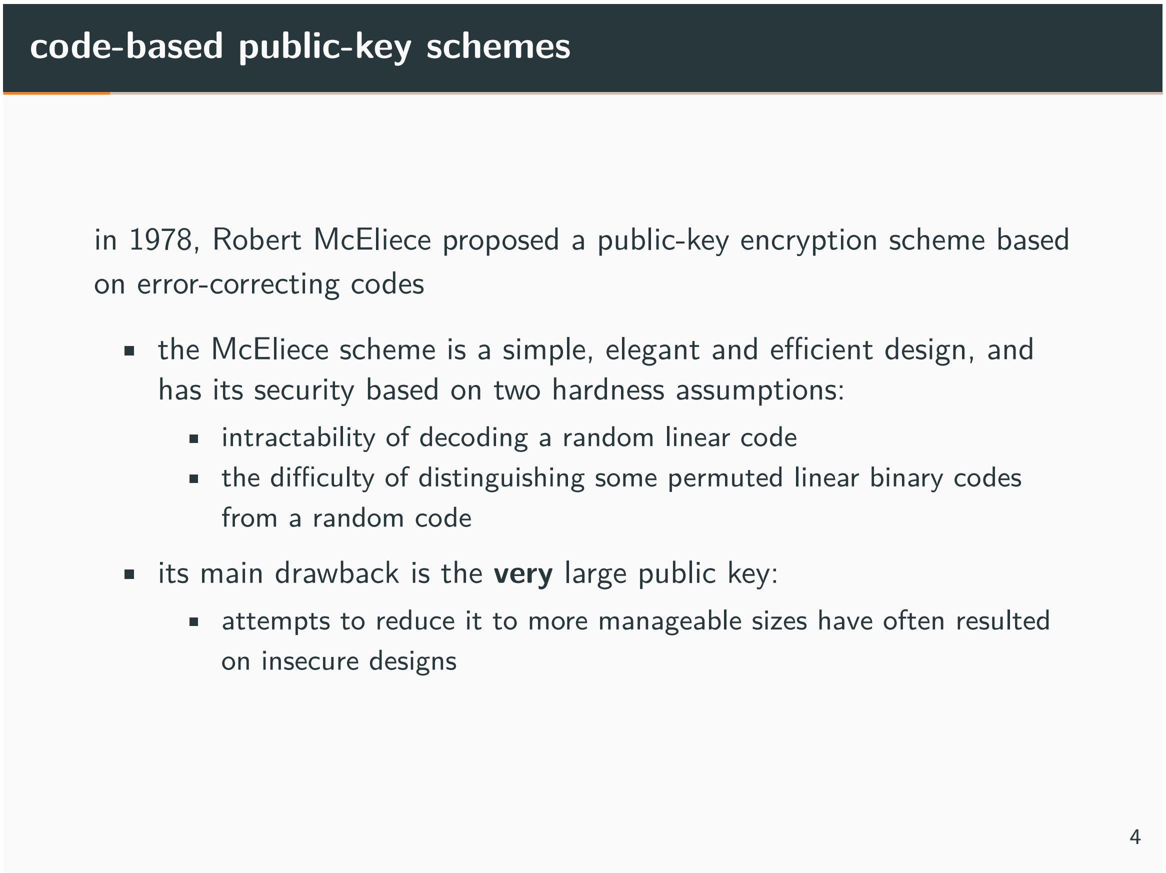
in this talk

we give an overview of code-based cryptography: public-key schemes that are based on error-correcting codes

- design and security: history and state-of-art
- code-based schemes in the NIST PQ competition



code-based pk schemes

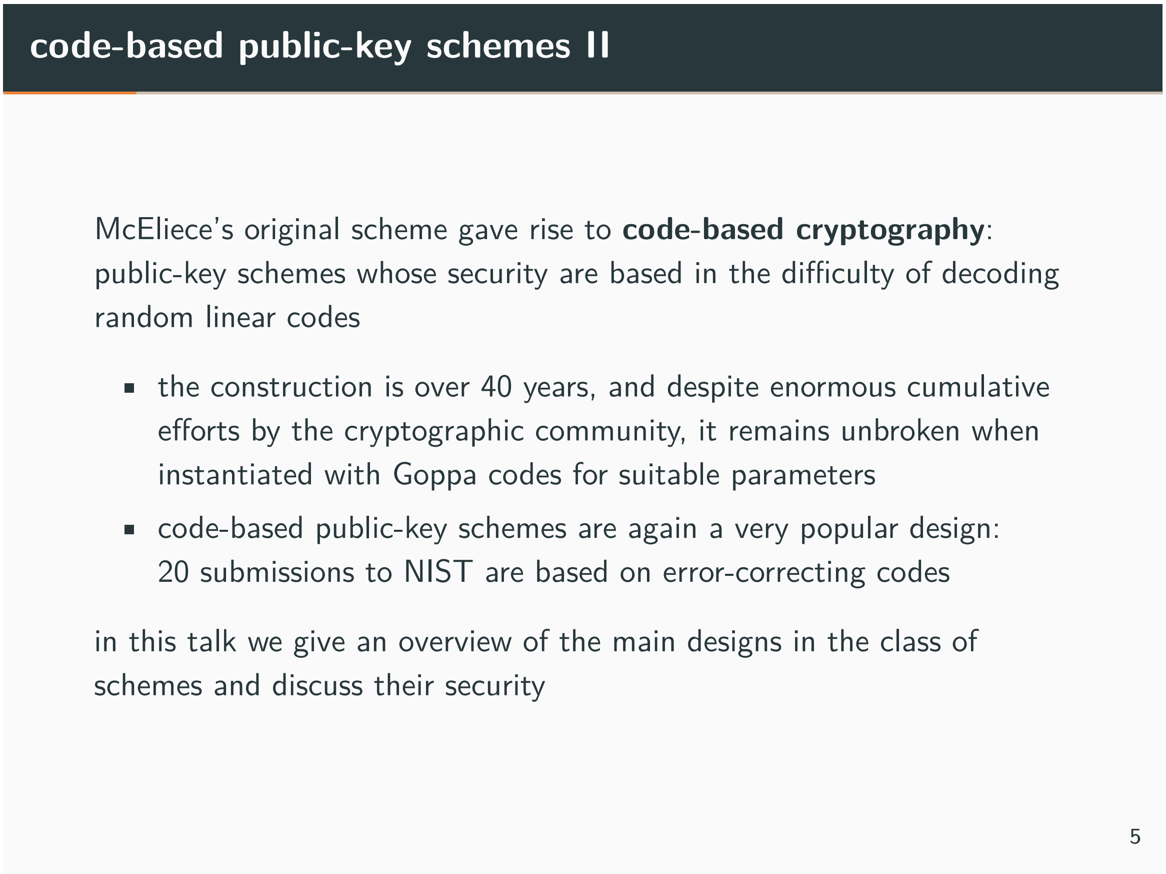


code-based public-key schemes

in 1978, Robert McEliece proposed a public-key encryption scheme based on error-correcting codes

- the McEliece scheme is a simple, elegant and efficient design, and has its security based on two hardness assumptions:
  - intractability of decoding a random linear code
  - the difficulty of distinguishing some permuted linear binary codes from a random code
- its main drawback is the **very** large public key:
  - attempts to reduce it to more manageable sizes have often resulted on insecure designs

4



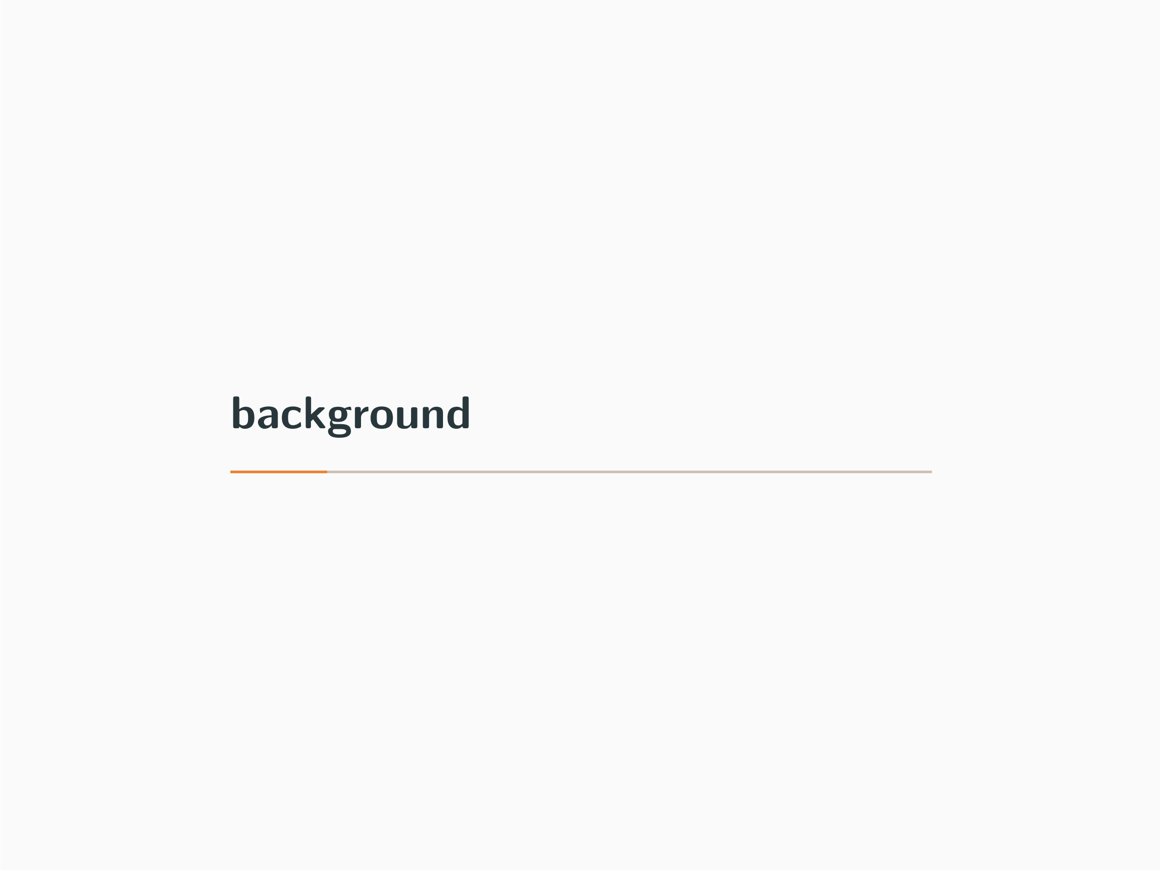
code-based public-key schemes II

McEliece's original scheme gave rise to **code-based cryptography**: public-key schemes whose security are based in the difficulty of decoding random linear codes

- the construction is over 40 years, and despite enormous cumulative efforts by the cryptographic community, it remains unbroken when instantiated with Goppa codes for suitable parameters
- code-based public-key schemes are again a very popular design: 20 submissions to NIST are based on error-correcting codes

in this talk we give an overview of the main designs in the class of schemes and discuss their security

5



background

error-correcting codes

process of "embedding" redundancy to messages, to allow for detection or correction of errors during transmission or storage.

$\mathcal{C} = [n, k, d]_2$: linear error-correcting code over $\mathbb{F}_2$ of length n and dimension k, with minimal distance d.

- $\mathcal{C}$ is capable of correcting at most $\tau = \left\lfloor \frac{d-1}{2} \right\rfloor$ errors
- $\mathcal{C}$ can be described by a generator matrix $\mathbf{G} \in \mathbb{F}_2^{k \times n}$, or a parity-check matrix $\mathbf{H} \in \mathbb{F}_2^{(n-k) \times n}$, such that $\mathbf{G} \cdot \mathbf{H}^T = \mathbf{0}$
- a vector $\mathbf{w} \in \mathbb{F}_2^k$ can be encoded as a codeword in $\mathcal{C}$ as

$$\mathbf{c} = \mathbf{w} \cdot \mathbf{G} \in \mathbb{F}_2^n$$

- for any codeword $\mathbf{c}$ in $\mathcal{C}$, we have $\mathbf{c} \cdot \mathbf{H}^T = \mathbf{0}$.

7

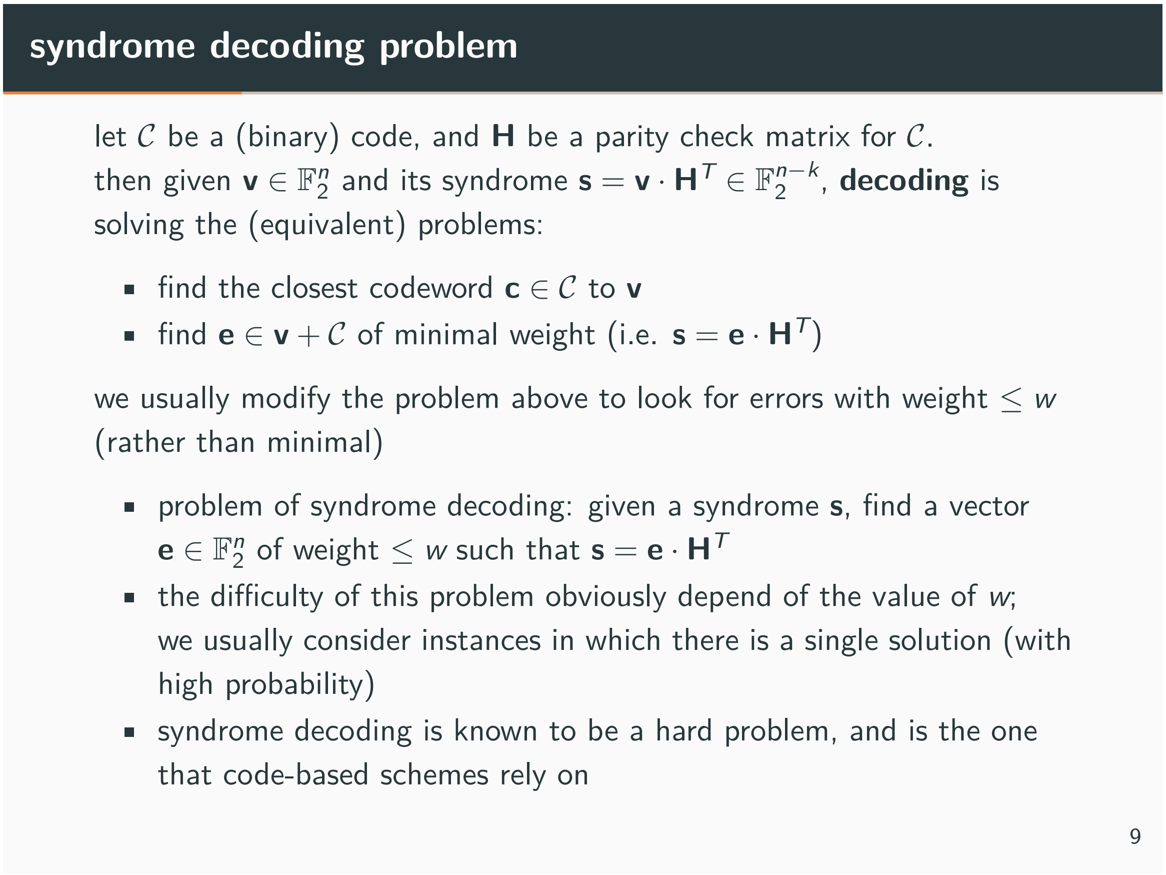
syndrome

for any $\mathbf{v} \in \mathbb{F}_2^n$, the vector $\mathbf{s} = \mathbf{v} \cdot \mathbf{H}^T \in \mathbb{F}_2^{n-k}$ is called a *syndrome*

- note that given a syndrome $\mathbf{s}$, and a parity check matrix $\mathbf{H}$, then finding a vector $\mathbf{v} \in \mathbb{F}_2^n$ such that $\mathbf{s} = \mathbf{v} \cdot \mathbf{H}^T$ is easy:
    - compute $\mathbf{H_0} = \mathbf{U} \cdot \mathbf{H} = [\mathbf{Id} | \mathbf{Q}]$
    - let $\mathbf{v} = (\mathbf{s} \cdot \mathbf{U}^T | 0)$
    - then we can show that $\mathbf{v} \cdot \mathbf{H}^T = \mathbf{s}$

however we are usually not interested in any coset representative $\mathbf{v}$, but rather one with small weight

8



syndrome decoding problem

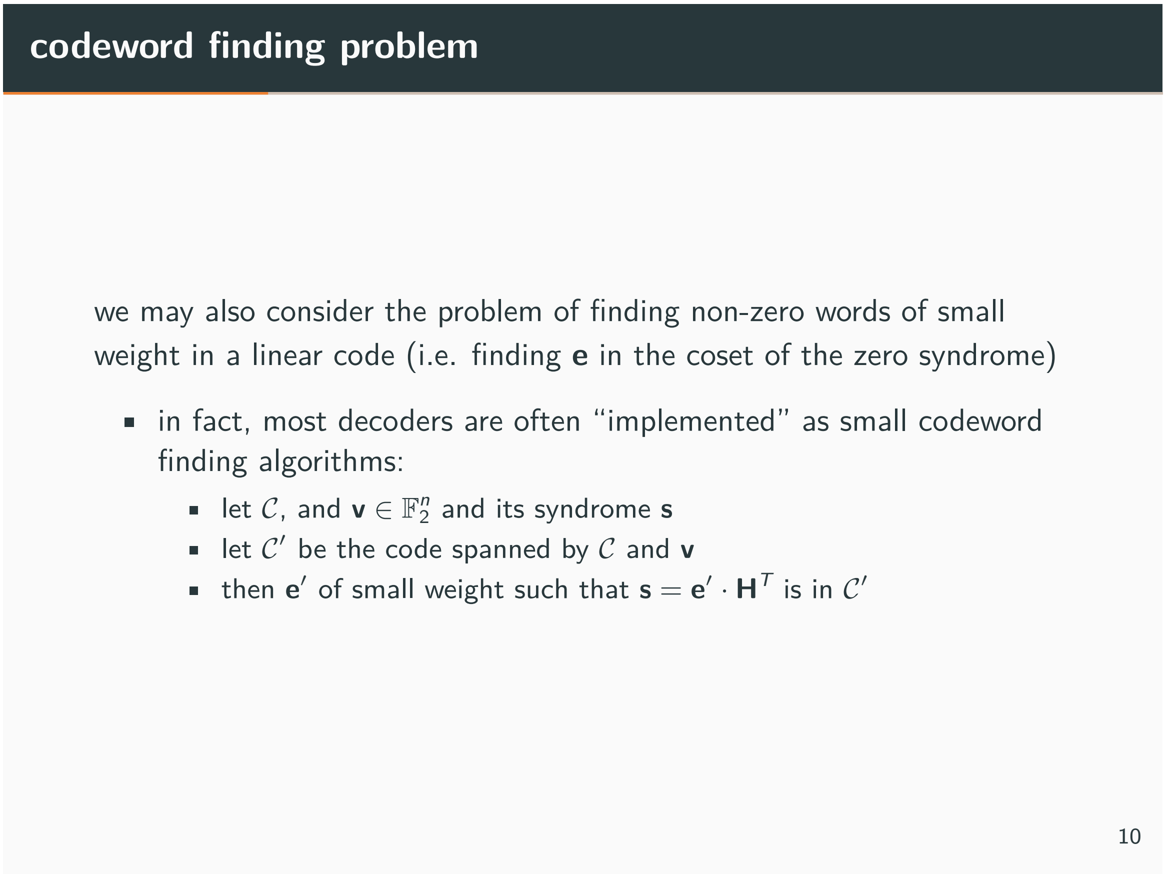
let $\mathcal{C}$ be a (binary) code, and $\mathbf{H}$ be a parity check matrix for $\mathcal{C}$.
then given $\mathbf{v} \in \mathbb{F}_2^n$ and its syndrome $\mathbf{s} = \mathbf{v} \cdot \mathbf{H}^T \in \mathbb{F}_2^{n-k}$, **decoding** is solving the (equivalent) problems:

- find the closest codeword $\mathbf{c} \in \mathcal{C}$ to $\mathbf{v}$
- find $\mathbf{e} \in \mathbf{v} + \mathcal{C}$ of minimal weight (i.e. $\mathbf{s} = \mathbf{e} \cdot \mathbf{H}^T$)

we usually modify the problem above to look for errors with weight $\leq w$ (rather than minimal)

- problem of syndrome decoding: given a syndrome $\mathbf{s}$, find a vector $\mathbf{e} \in \mathbb{F}_2^n$ of weight $\leq w$ such that $\mathbf{s} = \mathbf{e} \cdot \mathbf{H}^T$
- the difficulty of this problem obviously depend of the value of w; we usually consider instances in which there is a single solution (with high probability)
- syndrome decoding is known to be a hard problem, and is the one that code-based schemes rely on

9



codeword finding problem

we may also consider the problem of finding non-zero words of small weight in a linear code (i.e. finding $\mathbf{e}$ in the coset of the zero syndrome)

- in fact, most decoders are often "implemented" as small codeword finding algorithms:
  - let $\mathcal{C}$, and $\mathbf{v} \in \mathbb{F}_2^n$ and its syndrome $\mathbf{s}$
  - let $\mathcal{C}'$ be the code spanned by $\mathcal{C}$ and $\mathbf{v}$
  - then $\mathbf{e}'$ of small weight such that $\mathbf{s} = \mathbf{e}' \cdot \mathbf{H}^T$ is in $\mathcal{C}'$

10

Goppa codes

A binary separable Goppa code $\mathcal{C}_{\mathcal{G}}$ is a class of $[n, k, d]_2$ linear error-correcting codes defined by a Goppa polynomial $G(z) = g_0 + g_1 z + \cdots + g_\tau z^\tau \in \mathbb{F}_{2^m}[z]$ and $d = 2\tau + 1$, such that:

- $G(z)$ has no roots in $\mathbb{F}_{2^m}$, which implies $n = 2^m$
- $G(z)$ has no repeated roots in any extension field, which guarantees that $\mathcal{C}_{\mathcal{G}}$ is capable of correcting up to τ errors

Let $\mathbf{L} = (a_0, a_1, \ldots, a_{n-1})$ be a (ordered) sequence of all elements in $\mathbb{F}_{2^m}$. One can use $G(z)$ and $\mathbf{L}$ to construct its binary parity-check matrix and generator matrix

Goppa codes II

We write the parity-check matrix $\mathbf{H}_m \in \mathbb{F}_{2^m}^{\tau \times n}$ of $\mathcal{C}_{\mathcal{G}}$ using $G(z)$ and $\mathbf{L}$:

$$\mathbf{H}_m = \begin{bmatrix} G(a_0)^{-2} & G(a_1)^{-2} & G(a_2)^{-2} & \cdots & G(a_{n-1})^{-2} \\ a_0 G(a_0)^{-2} & a_1 G(a_1)^{-2} & a_2 G(a_2)^{-2} & \cdots & a_{n-1} G(a_{n-1})^{-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_0^{\tau-1} G(a_0)^{-2} & a_1^{\tau-1} G(a_1)^{-2} & a_2^{\tau-1} G(a_2)^{-2} & \cdots & a_{n-1}^{\tau-1} G(a_{n-1})^{-2} \end{bmatrix}$$

Let $B(a_i) = (b_{i0}, b_{i1}, \ldots, b_{i(m-1)})$ be a representation of a_i over $\mathbb{F}_2$:

$$a_i = b_{i0} + b_{i1}\beta + b_{i2}\beta^2 + \cdots + b_{i(m-1)}\beta^{m-1}$$

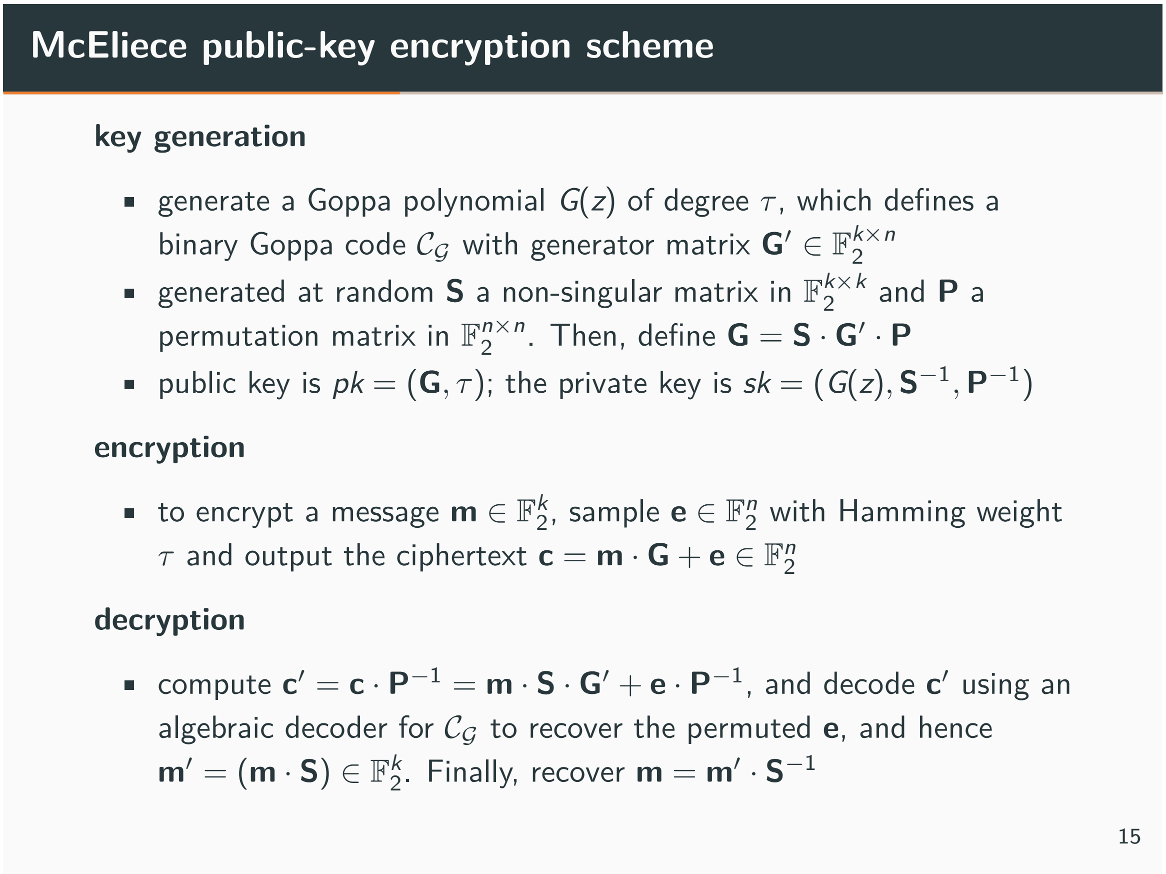
Then by replacing each entry of $\mathbf{H}_m$ with $B(\cdot)^T$, we have the binary parity-check matrix $\mathbf{H} \in \mathbb{F}_2^{m\tau \times n}$, which has rank $m\tau = n - k$.
The generator matrix $\mathbf{G} \in \mathbb{F}_2^{k \times n}$ can then be easily obtained from $\mathbf{H}$.

Goppa codes III

Binary Goppa codes are algebraic codes with high error correction capability in relation to the code rate (k/n):

- with knowledge of its structure, binary Goppa codes can be efficiently decoded by using Patterson's method or the Berlekamp-Massey algorithm
- however if the code structure is *hidden*, then decoding binary Goppa codes is *expected* to be as hard as decoding a random linear code
    - in fact, it is conjectured to be "indistinguishable" from random linear codes (you cannot tell generator matrices apart)...
    - ... and the best *currently known* algorithms are based on the technique known as *information-set decoding*, originally proposed by Prange

McEliece PKE scheme



McEliece public-key encryption scheme

key generation

- generate a Goppa polynomial $G(z)$ of degree τ, which defines a binary Goppa code $\mathcal{C}_{\mathcal{G}}$ with generator matrix $\mathbf{G}' \in \mathbb{F}_2^{k \times n}$
- generated at random $\mathbf{S}$ a non-singular matrix in $\mathbb{F}_2^{k \times k}$ and $\mathbf{P}$ a permutation matrix in $\mathbb{F}_2^{n \times n}$. Then, define $\mathbf{G} = \mathbf{S} \cdot \mathbf{G}' \cdot \mathbf{P}$
- public key is $pk = (\mathbf{G}, \tau)$; the private key is $sk = (G(z), \mathbf{S}^{-1}, \mathbf{P}^{-1})$

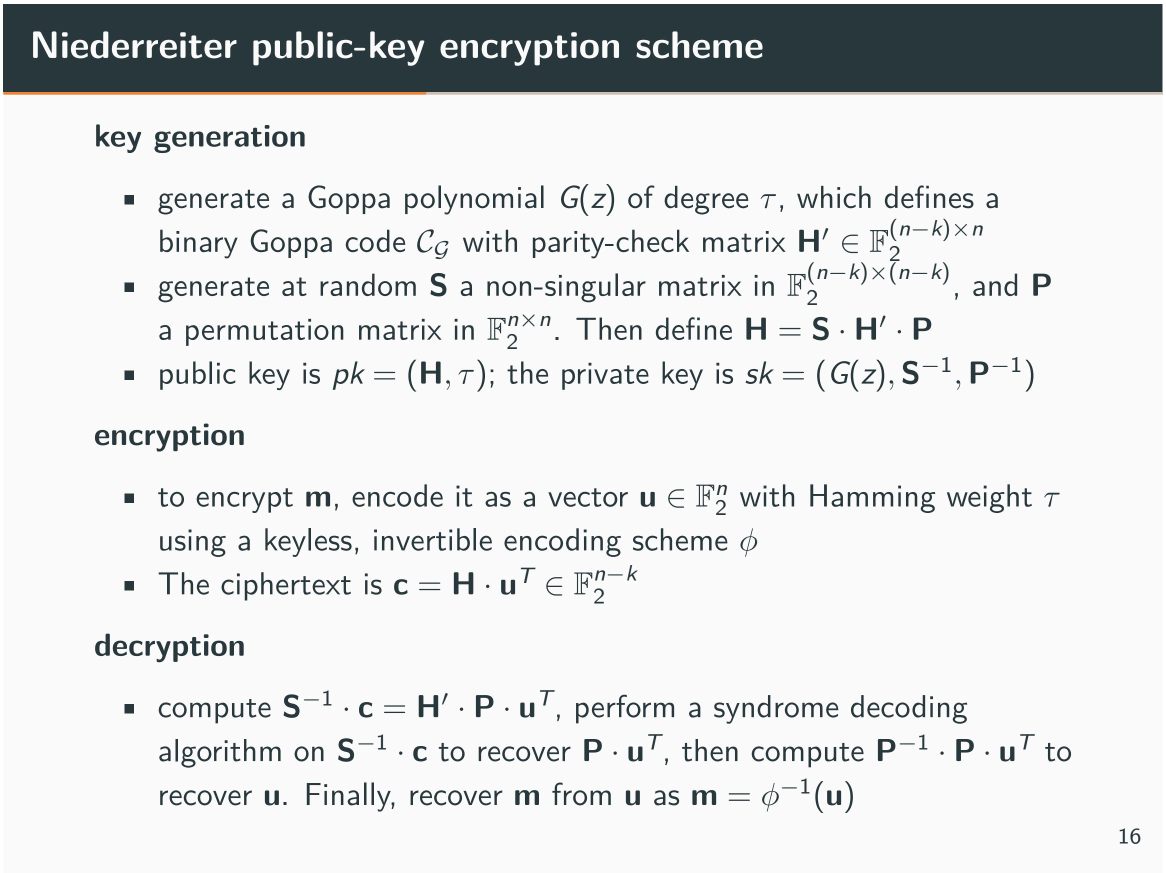
encryption

- to encrypt a message $\mathbf{m} \in \mathbb{F}_2^k$, sample $\mathbf{e} \in \mathbb{F}_2^n$ with Hamming weight τ and output the ciphertext $\mathbf{c} = \mathbf{m} \cdot \mathbf{G} + \mathbf{e} \in \mathbb{F}_2^n$

decryption

- compute $\mathbf{c}' = \mathbf{c} \cdot \mathbf{P}^{-1} = \mathbf{m} \cdot \mathbf{S} \cdot \mathbf{G}' + \mathbf{e} \cdot \mathbf{P}^{-1}$, and decode $\mathbf{c}'$ using an algebraic decoder for $\mathcal{C}_{\mathcal{G}}$ to recover the permuted $\mathbf{e}$, and hence $\mathbf{m}' = (\mathbf{m} \cdot \mathbf{S}) \in \mathbb{F}_2^k$. Finally, recover $\mathbf{m} = \mathbf{m}' \cdot \mathbf{S}^{-1}$

15



Niederreiter public-key encryption scheme

key generation

- generate a Goppa polynomial $G(z)$ of degree τ, which defines a binary Goppa code $\mathcal{C}_{\mathcal{G}}$ with parity-check matrix $\mathbf{H}' \in \mathbb{F}_2^{(n-k) \times n}$
- generate at random $\mathbf{S}$ a non-singular matrix in $\mathbb{F}_2^{(n-k) \times (n-k)}$, and $\mathbf{P}$ a permutation matrix in $\mathbb{F}_2^{n \times n}$. Then define $\mathbf{H} = \mathbf{S} \cdot \mathbf{H}' \cdot \mathbf{P}$
- public key is $pk = (\mathbf{H}, \tau)$; the private key is $sk = (G(z), \mathbf{S}^{-1}, \mathbf{P}^{-1})$

encryption

- to encrypt $\mathbf{m}$, encode it as a vector $\mathbf{u} \in \mathbb{F}_2^n$ with Hamming weight τ using a keyless, invertible encoding scheme ϕ
- The ciphertext is $\mathbf{c} = \mathbf{H} \cdot \mathbf{u}^T \in \mathbb{F}_2^{n-k}$

decryption

- compute $\mathbf{S}^{-1} \cdot \mathbf{c} = \mathbf{H}' \cdot \mathbf{P} \cdot \mathbf{u}^T$, perform a syndrome decoding algorithm on $\mathbf{S}^{-1} \cdot \mathbf{c}$ to recover $\mathbf{P} \cdot \mathbf{u}^T$, then compute $\mathbf{P}^{-1} \cdot \mathbf{P} \cdot \mathbf{u}^T$ to recover $\mathbf{u}$. Finally, recover $\mathbf{m}$ from $\mathbf{u}$ as $\mathbf{m} = \phi^{-1}(\mathbf{u})$

16

McEliece & Niederreiter schemes

a few remarks:

- McEliece: message is encoded by $\mathcal{C}_{\mathcal{G}}$, and random errors are added to produce the ciphertext;
  Niederreiter: message is encoded as a low-weight vector $\mathbf{u}$, and the ciphertext is represented as the syndrome of $\mathbf{u}$
- security based on two hardness assumptions:
  - intractability of decoding a random linear code
  - difficulty of distinguishing permuted Goppa code from random code
- note that, as presented, they are neither IND-CPA or IND-CCA secure
  - they satisfiy a weaker notion of security: one-wayness (OW)

McEliece & Niederreiter schemes II

a few remarks:

- Niederreiter's scheme was originally proposed with Reed-Solomon codes (and subsequently broken)
  - when using Goppa codes, it is known that the security of the Niederreiter and the McEliece schemes are equivalent
- the schemes have efficient operation (particularly encryption) and compact ciphertexts.
- public key is however **very** large! (a matrix in $\mathbb{F}_2^{k \times n}$)
  - for example, for $n = 8192, k = 7815, \tau = 29$, offering $\sim$ 128-bit (classical) security, the public key has $\sim$ 360KB
  - for $\sim$ 256-bit security, we are looking at $\sim$ 1MB public keys
  - note that Niederreiter's scheme allows for a reduction of the public key: $\mathbf{H}$ can be given in "systematic form"; however many attempts to further "compress" the public key have led to insecure designs

code-based vs lattice-based

note a relationship between code-based and lattice-based crypto

- lattice-based schemes try to hide a secret vector in a high-dimensional lattice **over q** by introducing small errors to **all coordinates**
- code-based schemes try to hide a secret vector in a **very** high-dimensional lattice **over 2** by introducing errors to **some coordinates**

code-based crypto: security

security

possible attacks:

1. message recovery: try to determine the error vector from the ciphertext and the public key
2. key recovery: try to recover the original structure of the code $\mathcal{C}_{\mathcal{G}}$

one may also perform chosen-ciphertext attacks, exploring the lack of IND-CPA and IND-CCA security (it may be addressed by turning McEliece into a IND-CCA secure PKE scheme)

McEliece scheme: all known attacks for (2) are less efficient than (1)

- information-set decoding (ISD): originally proposed by Prange in the 1960s, is the best message-recovery attack against McEliece

information-set decoding: basic idea

let $\mathbf{c} = \mathbf{m} \cdot \mathbf{G} + \mathbf{e} \in \mathbb{F}_2^n$ be a McEliece ciphertext of a message $\mathbf{m} \in \mathbb{F}_2^k$, where $\mathbf{e} \in \mathbb{F}_2^n$ has Hamming weight τ

1. assume $\mathbf{G} = [\mathbf{M}|\mathbf{Q}]$, where $\mathbf{M} \in \mathbb{F}_2^{k \times k}$ is invertible
2. assume that the first k entries of $\mathbf{c}$ are error-free

then we can recover $\mathbf{m}$:

- if $\mathbf{c}' \in \mathbb{F}_2^k$ is the vector of the first k entries of $\mathbf{c}$, then it is clear that $\mathbf{m} = \mathbf{c}' \cdot \mathbf{M}^{-1}$
- we can also recover $\mathbf{e} = \mathbf{c} + \mathbf{m} \cdot \mathbf{G}$

how can we check whether assumption (2) is valid?

- compute $\mathbf{m}' = \mathbf{c}' \cdot \mathbf{M}^{-1}$, and check whether $\mathbf{e}' = \mathbf{c} + \mathbf{m}' \cdot \mathbf{G}$ has Hamming weight τ

information-set decoding

Prange's basic ISD algorithm works as:

1. randomly select k columns of the generator matrix $\mathbf{G}$ (an "information set")
2. if $\mathbf{M}$ is not invertible, go to 1.
3. compute $\mathbf{e}' = \mathbf{c} + (\mathbf{c}' \cdot \mathbf{M}^{-1}) \cdot \mathbf{G}$
4. if $\mathbf{e}'$ has Hamming weight τ, return $\mathbf{e}'$; otherwise go to 1

for an information-set attack to work, the selected k bits of the ciphertext need to be free from error;
one can work out this probability (and that the resulting $k \times k$ matrix is invertible) to derive the complexity of the (basic) ISD attack against McEliece

information-set decoding II

several improvements have followed from the basic ISD algorithm proposed by Prange

- Leon, Lee-Brickel, Stern, Canteaut-Chabaud, Bernstein-Lange-Peters, Finiasz-Sendrier, Becker-Joux-May-Meurer, May-Ozerov,...
- most modern ISD attacks are based on collisions between the calculated syndrome of the target ciphertext and syndromes calculated from selected columns of the parity-check matrix.
- no closed form formula for complexity given parameter set – rather one can estimate total cost as binary work factor
- recent work have resulted on improved asymptotic complexity for syndrome decoding algorithms

information-set decoding III

quantum attacks?

- currently, there are no known *dedicated* quantum algorithms for attacking code-based schemes
- best approach to exploit quantum computers to attack code-based schemes is the application of generic quantum techniques (e.g. Grover's algorithm) to speed up ISD
- recent work indicates that quantum attacks will *at best* offer a square-root reduction in the classical security levels

structural attacks: private-key recovery

so far we have considered message recovery attacks under the assumption that the structure of the (secret) code cannot be recovered from the public key

- in this case, we assume the best attacks are based on generic decoding algorithms for random linear codes
- however McEliece public keys are not random codes, but rather an algebraic code (Goppa code) for which its structure has been hidden (via the permutation of its columns).
- security of McEliece PKE relies on the secrecy of the permutation matrix $\mathbf{P}$

Structural attacks: attacks which attempt to recover the original code structure from the public key

- for Goppa codes, the best known attack for recovery of the original code is essentially an exhaustive search for irreducible Goppa polynomials $G(z)$

codes for code-based crypto

to be used in code-based cryptography, you want:

- good error-correcting capability and efficient decoding
- to be able to *hide* the code (algebraic) structure via a secret isometry (in the case of binary linear codes, equivalent to a permutation of its support)
  - that it is hard to recover the original structure from the generator matrix of the permuted code
  - that the generator matrix of the permuted code is indistinguishable from the generator matrix of a random code

currently, binary Goppa codes are the only class of codes that we know that satisfies these properties; it however results on very large codes.

- search for alternatives have often failed: Generalised Reed-Solomon codes, Reed-Muller codes, rank metric codes, (quasi) cyclic codes,...

NTS-KEM: a NIST submission

NTS-KEM: a code-based KEM scheme

NTS-KEM: a IND-CCA, code-based KEM scheme, submitted to NIST (submitters: Tjhai, Tomlinson, Albrecht, Cid and Paterson)

- a variant of the McEliece and Niederreiter schemes
- proven to be secure against chosen ciphertext attacks, under the assumption that inverting the McEliece pk encryption is hard
- ciphertexts are relatively compact, making the scheme suitable for applications with limited communication bandwidth.
- conservative choice of code and parameters (leading to large public keys)
- main goal: offer long-term post-quantum security for suitable applications

NTS-KEM: operations I

key generation is similar to McEliece, but with generator matrix $\mathbf{G} = [\mathbf{I}|\mathbf{Q}]$ in systematic form for public key (and some reduction in the private key)

encapsulation

1. generate random error vector $\mathbf{e} \in \mathbb{F}_2^n$ with Hamming weight τ
2. partition $\mathbf{e} = (\mathbf{e}_a \mid \mathbf{e}_b \mid \mathbf{e}_c)$, where $\mathbf{e}_a \in \mathbb{F}_2^{k-\ell}$, $\mathbf{e}_b \in \mathbb{F}_2^{\ell}$ and $\mathbf{e}_c \in \mathbb{F}_2^{n-k}$.
3. compute $\mathbf{k}_e = H_\ell(\mathbf{e}) \in \mathbb{F}_2^{\ell}$.
4. construct the message vector $\mathbf{m} = (\mathbf{e}_a \mid \mathbf{k}_e) \in \mathbb{F}_2^{k}$.
5. perform systematic encoding of $\mathbf{m}$ with $\mathbf{Q}$:
$$\begin{aligned} \mathbf{c} &= (\mathbf{m} \mid \mathbf{m} \cdot \mathbf{Q}) + \mathbf{e} \\ &= (\mathbf{e}_a \mid \mathbf{k}_e \mid (\mathbf{e}_a \mid \mathbf{k}_e) \cdot \mathbf{Q}) + (\mathbf{e}_a \mid \mathbf{e}_b \mid \mathbf{e}_c) \\ &= (\mathbf{0}_a \mid \mathbf{k}_e + \mathbf{e}_b \mid (\mathbf{e}_a \mid \mathbf{k}_e) \cdot \mathbf{Q} + \mathbf{e}_c) \\ &= (\mathbf{0}_a \mid \mathbf{c}_b \mid \mathbf{c}_c)\,, \end{aligned}$$
where $\mathbf{c}_b = \mathbf{k}_e + \mathbf{e}_b$ and $\mathbf{c}_c = (\mathbf{e}_a \mid \mathbf{k}_e) \cdot \mathbf{Q} + \mathbf{e}_c$;
remove the first $k - \ell$ coordinates from $\mathbf{c}$ to obtain $\mathbf{c}^* = (\mathbf{c}_b \mid \mathbf{c}_c) \in \mathbb{F}_2^{n-k+\ell}$.
6. output the pair $(\mathbf{k}_r, \mathbf{c}^*)$ where $\mathbf{k}_r = H_\ell(\mathbf{k}_e \mid \mathbf{e}) \in \mathbb{F}_2^{\ell}$.

NTS-KEM: operations II

decapsulation: given ciphertext $\mathbf{c}^* = (\mathbf{c}_b \mid \mathbf{c}_c)$:

1. let $\mathbf{c} = (\mathbf{0}_a \mid \mathbf{c}_b \mid \mathbf{c}_c) \in \mathbb{F}_2^n$, and apply a decoding algorithm — using the secret key — to recover a permuted error pattern $\mathbf{e}'$.
2. compute the error vector $\mathbf{e} = \pi_{\mathbf{p}}(\mathbf{e}')$.
3. partition $\mathbf{e} = (\mathbf{e}_a \mid \mathbf{e}_b \mid \mathbf{e}_c)$, where $\mathbf{e}_a \in \mathbb{F}_2^{k-\ell}$, $\mathbf{e}_b \in \mathbb{F}_2^{\ell}$ and $\mathbf{e}_c \in \mathbb{F}_2^{n-k}$, and compute $\mathbf{k}_e = \mathbf{c}_b - \mathbf{e}_b$.
4. verify that $H_\ell(\mathbf{e}) = \mathbf{k}_e$ and $\mathrm{hw}(\mathbf{e}) = \tau$.
   - if yes, return $\mathbf{k}_r = H_\ell(\mathbf{k}_e \mid \mathbf{e}) \in \mathbb{F}_2^{\ell}$; otherwise return $\perp$.

NTS-KEM: parameters

parameters

algorithm version	security category	security target	n	k	d	τ	pk size (in bytes)	sk size (in bytes)	ctext size (in bits)
NTS-KEM(12,64)	1	128-bit	4096	3328	129	64	319,488	9,216	1,024
NTS-KEM(13,80)	3	192-bit	8192	7152	161	80	929,760	17,524	1,296
NTS-KEM(13,136)	5	256-bit	8192	6424	273	136	1,419,704	19,890	2,024

NTS-KEM: features

- conservative design, with strong security guarantees and conservative parameter sets
- security based on a simple and well-understood mathematical problem
- scheme with high degree of flexibility in the setting of parameters
- private-public key pairs may be deployed for long periods of time
- compact ciphertexts, around 2,000 bits at the highest security level.
- efficient operations, particularly encapsulation, leading to reasonably fast software implementations.
  - simplicity of the operations and subroutines allow for the straightforward deployment of protection measures against side-channel attacks

notable disadvantage: size of the public key

- at the highest security level proposed, the public key is approximately 1.39MB in size (312KB for the 128-bit security version).

33

NIST PQ competition

NIST PQ standardization process

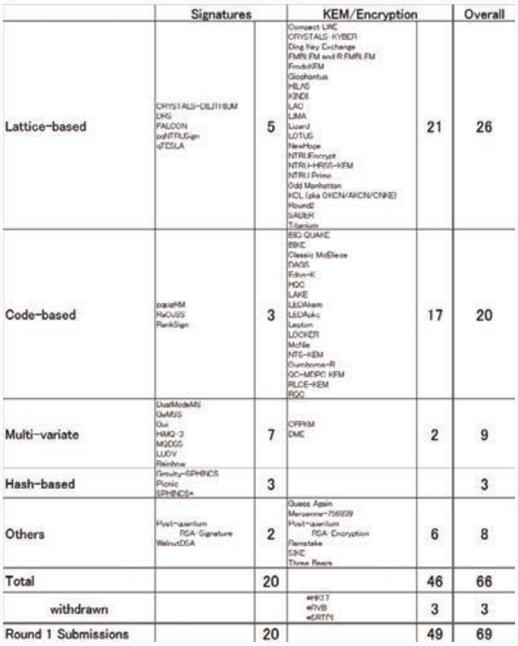
not a “competition” but a “*process to solicit, evaluate, and standardize one or more quantum-resistant public-key cryptographic algorithms*”:

- to eventually replace the NIST pk standards for signature (FIPS 186) and key establishment (SP 800-56A/B)
- call for quantum-resistant *signature*, *pk encryption* and *key-establishment mechanisms (KEM)* schemes.
- main criteria: security (classic/quantum attacks, different security levels) and performance (mainly software)... plus other desirable features (e.g. simplicity and flexibility, suitability to existing/new applications, resistance against side-channel attacks, etc).
- timeline:
  - *Feb 2016*: NIST official announcement at PQCrypto 2016 in Fukuoka
  - *Dec 2016*: call for proposals
  - *30 Nov 2017*: deadline for submissions
  - *3-5 years*: analysis phase, in 2-3 phases, resulting on official reports, events and narrowing of pool of candidates
  - *mid-2020s*: draft standards available for public comments

35

NIST PQ standardization process

- 30 Dec 2017: **82** submissions received (23 signature; 59 PKE/KEM)
- 2 submissions were withdrawn, leading to **80** (=23+57) schemes
- 21 Dec 2017: NIST announces the **69** (=20+49) ”complete and proper” submissions to be considered in the process
- soon after, attacks led to 3 submissions being officially withdrawn (SRTPI, HK17, RVB)



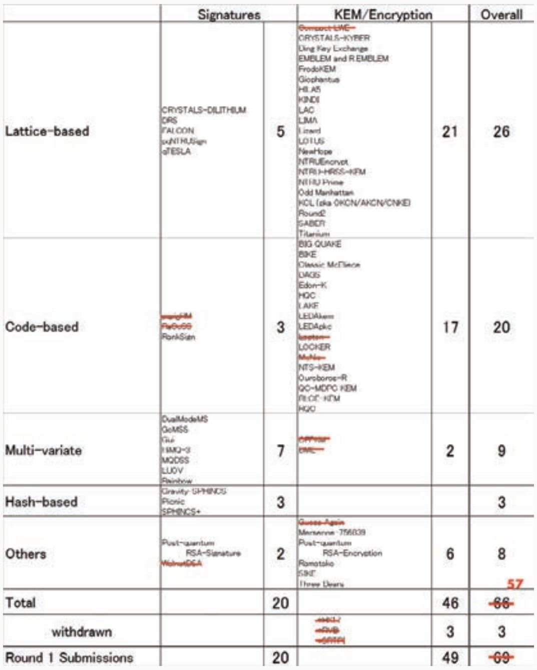
	Signatures		KEM/Encryption		Overall
Lattice-based	CRYSTALS-DILITHIUM DRS FALCON pqNTRUSign qTESLA	5	Compact LWE CRYSTALS-KYBER Ding Key Exchange EMBLEM and R.EMBLEM FrodoKEM Giophantus HILA5 KINDI LAC LIMA Lizard LOTUS NewHope NTRUEncrypt NTRU-HRSS-KEM NTRU Prime Odd Manhattan KCL (aka OKCN/AKCN/CNKE) Round2 SABER Titanium	21	26
Code-based	pqsigRM RaCoSS RankSign	3	BIG QUAKE BIKE Classic McEliece DAGS Edon-K HQC LAKE LEDAkem LEDApkc Lepton LOCKER McNie NTS-KEM Ouroboros-R QC-MDPC KEM RLCE-KEM RQC	17	20
Multi-variate	DualModeMS GeMSS Gui HiMQ-3 MQDSS LUOV Rainbow	7	CFPKM DME	2	9
Hash-based	Gravity-SPHINCS Picnic SPHINCS+	3			3
Others	Post-quantum RSA-Signature WalnutDSA	2	Guess Again Mersenne-756839 Post-quantum RSA-Encryption Ramstake SIKE Three Bears	6	8
Total		20		46	66
withdrawn			*HK17 *RVB *SRTPI	3	3
Round 1 Submissions		20		49	69

* table due to Ryo Fujita.

36

NIST PQ standardization process

- 30 Dec 2017: **82** submissions received (23 signature; 59 PKE/KEM)
- 2 submissions were withdrawn, leading to **80** (=23+57) schemes
- 21 Dec 2017: NIST announces the **69** (=20+49) "complete and proper" submissions to be considered in the process
- soon after, attacks led to 3 submissions being officially withdrawn (SRTPI, HK17, RVB)
- another 9 are considered either broken or seriously wounded



	Signatures		KEM/Encryption		Overall
Lattice-based	CRYSTALS-DILITHIUM DRS FALCON pqNTRUSign qTESLA	5	~~Compact LWE~~ CRYSTALS-KYBER Ding Key Exchange EMBLEM and R.EMBLEM FrodoKEM Giophantus HILA5 KINDI LAC LIMA Lizard LOTUS NewHope NTRUEncrypt NTRU-HRSS-KEM NTRU Prime Odd Manhattan KCL (aka OKCN/AKCN/CNKE) Round2 SABER Titanium	21	26
Code-based	~~pqsigRM~~ ~~RaCoSS~~ RankSign	3	BIG QUAKE BIKE Classic McEliece DAGS Edon-K HQC LAKE LEDAkem LEDApkc ~~Lepton~~ LOCKER ~~McNie~~ NTS-KEM Ouroboros-R QC-MDPC KEM RLCE-KEM HQC	17	20
Multi-variate	DualModeMS GeMSS Gui HiMQ-3 MQDSS LUOV Rainbow	7	~~CFPKM~~ ~~DME~~	2	9
Hash-based	Gravity-SPHINCS Picnic SPHINCS+	3			3
Others	Post-quantum RSA-Signature ~~WalnutDSA~~	2	~~Guess Again~~ Mersenne-756839 Post-quantum RSA-Encryption Ramstake SIKE Three Bears	6	8
Total		20		46	57 ~~66~~
withdrawn			~~HK17~~ ~~RVB~~ ~~SRTPI~~	3	3
Round 1 Submissions		20		49	~~69~~

* table due to Ryo Fujita.

NIST PQ standardization process

NIST PQ process is a very challenging one:

- call for proposals for three primitives (signature, pke and kem), with several applications and possible trade-offs
- very large number of submissions:
  - AES (1997-2001): 15 submissions
  - eStream (2004-2008): 34 submissions
  - SHA-3 (2007-2012): 51 submissions
  - CAESAR (2013-): 57 submissions
- transition and migration also very challenging, may take 10 years (potentially to "hybrid mode" deployment)

code-based schemes in NIST PQSP

NIST is considering 17 code-based submissions
(16 KEM/PKE + 1 signature)

- common approaches to reduce public-key sizes: based on cyclic/quasi-cyclic structures and/or reduced-density codes (e.g. MDPC codes), use of rank metric, etc
- some issues: efficiency of decoding; attacks based on decoding failures; key-recovery security
- many of these approaches have led to insecure designs in the past

for over 40 years of research in code-based cryptography, only Goppa code based schemes have resisted attacks and shown to be secure...

NIST code-based submissions: examples

Classic McEliece (by Bernstein et al.)

- modern version of Niederreiter's scheme (using Goppa codes and offering IND-CCA security)
- two versions (both cat 5, claiming 256-bit classic security):
    - KEM with $m = 13$, $n = 6960$, and $\tau = 119$
        - 1,047,319-byte public keys, 13,908-byte private keys, 226-byte ciphertexts, and 32-byte session keys
    - KEM with $m = 13$, $n = 8192$, and $\tau = 128$
        - 1,357,824-byte public keys, 14,080-byte private keys, 240-byte ciphertexts, and 32-byte session keys

comparison: for cat 5, NTS-KEM has $m = 13$, $n = 8192$, and $\tau = 136$, leading to 1,419,704-byte public keys, 19,890-byte private keys, 253-byte ciphertexts, and 32-byte session keys

- but pk and ciphertext sizes would be the same if using $\tau = 128$ (NTS-KEM sk would be longer)

NIST code-based submissions: examples

BIKE: Bit Flipping Key Encapsulation (Aragon et al.)

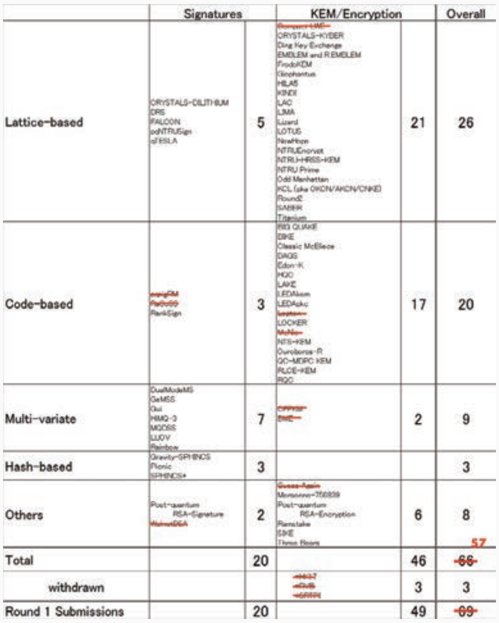
- KEM based on quasi-cyclic moderate density parity-check (QC-MDPC) codes that can be decoded using bit flipping decoding techniques
  - QC codes have block-circulant matrices as generator matrix
  - MDPC (Moderate Density Parity Check) codes admit a somewhat sparse parity check matrix; allowing for the use of iterative decoders (in particular, bit flipping decoders)
- three versions (based on McEliece, Niederreiter, Ouroboros)
- use of QC codes lead to more compact public keys (though larger parameters)
- early version was broken via a reaction attack in 2016
  - iterative decoding can fail with some small probability, and one may identify a dependence between the secret key and the failure in decoding
- as a result, BIKE now uses ephemeral KEM key pairs

NIST code-based submissions: examples

- LAKE (Aragon et al.): Ideal-LRPC (Low Rank Parity Check) codes, using rank metric
- RLCE (Wang): generalised Reed-Solomon code with a number of random parity check columns appended
- etc

final thoughts

- code-based schemes & NIST PQ standardisation process



	Signatures		KEM/Encryption		Overall
Lattice-based	CRYSTALS-DILITHIUM DRS FALCON pqNTRUSign qTESLA	5	~~Compact LWE~~ CRYSTALS-KYBER Ding Key Exchange EMBLEM and R.EMBLEM FrodoKEM Giophantus HILA5 KINDI LAC LIMA Lizard LOTUS NewHope NTRUEncrypt NTRU-HRSS-KEM NTRU Prime Odd Manhattan KCL (aka OKCN/AKCN/CNKE) Round2 SABER Titanium	21	26
Code-based	~~pqsigRM~~ ~~RaCoSS~~ RankSign	3	BIG QUAKE BIKE Classic McEliece DAGS Edon-K HQC LAKE LEDAkem LEDApkc ~~Lepton~~ LOCKER ~~McNie~~ NTS-KEM Ouroboros-R QC-MDPC KEM RLCE-KEM RQC	17	20
Multi-variate	DualModeMS GeMSS Gui HiMQ-3 MQDSS LUOV Rainbow	7	~~CFPKM~~ ~~DME~~	2	9
Hash-based	Gravity-SPHINCS Picnic SPHINCS+	3			3
Others	Post-quantum RSA-Signature ~~WalnutDSA~~	2	~~Guess Again~~ Mersenne-756839 Post-quantum RSA-Encryption Ramstake SIKE Three Bears	6	8
Total		20		46	57 ~~66~~
withdrawn			~~HK17~~ ~~CVB~~ ~~SRTPI~~	3	3
Round 1 Submissions		20		49	~~69~~

thank you ... questions?

Workshop on analysis of mathematical cryptography via algebraic methods
@Nishijin Praza, Fukuoka, Japan

Solving RSA and factoring problems using LLL reduction

Atsushi Takayasu
The University of Tokyo, AIST

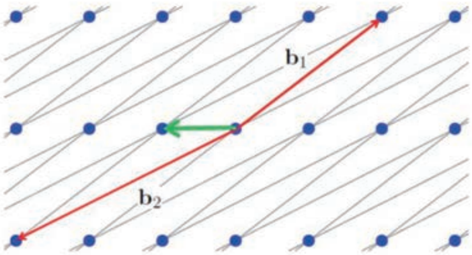
2018/2/6

Today’s talk

➢ *Coppersmith’s methods*
Constructing polynomial time algorithms for solving integer/modular equations with small roots using the LLL lattice basis reduction.
Basic applications: Solving RSA and factorization problems

Today’s talk

- Basic concepts
- Overview of the methods
- Recent results [TLP@EC’17]



Background

RSA

Public Key: $(N = pq, e)$ Secret Key: (p, q, d)

$$ed = 1 \mod (p-1)(q-1)$$

- Encryption of $m \in \mathbb{Z}^*_{(p-1)(q-1)}$

$$c = m^e \mod N$$

- Decryption of $c \in \mathbb{Z}^*_{(p-1)(q-1)}$

$$m = c^d \mod N$$

How about its security?

Factoring attack

Public Key: $(N = pq, e)$ Secret Key: (p, q, d)

$$ed = 1 \mod (p-1)(q-1)$$

$$c = m^e \mod N \qquad m = c^d \mod N$$

If we can efficiently solve a bivariate integer equation

$$f(x, y) = xy - N = 0,$$

we can recover all secret (p, q, d).

It seems infeasible for large *N*.

5/55

Plaintext recovery attack

Public Key: $(N = pq, e)$ Secret Key: (p, q, d)

$$ed = 1 \mod (p-1)(q-1)$$

$$c = m^e \mod N \qquad m = c^d \mod N$$

If we can efficiently solve a univariate modular equation

$$f(x) = x^e - c = 0 \mod N,$$

we can recover a plaintext m.

It seems infeasible when the factorization of *N* is hard.

6/55

Coppersmith's methods

In [Cop@EC'96a], [Cop@EC'96b], Coppersmith partially resolved the problem.

Coppersmith methods solve integer/modular equations with *small roots* in polynomial time.

Factoring attack with hint

In general, it seems computationally infeasible to solve

$$f(x,y) = xy - N = 0$$

and recover (p, q).

What happens if we get the most significant bits (p', q')?

Coppersmith's methods can solve an integer equation

$$f(x,y) = (p'+x)(q'+y) - N = 0$$

if $|p-p'| < N^{1/4}, |q-q'| < N^{1/4}$.

Plaintext recovery attack with hint

In general, it seems computationally infeasible to solve

$$f(x) = x^e - c = 0 \mod N$$

and recover m.

What happens if we get the most significant bits m' ?

Coppersmith's methods can solve an integer equation

$$f(x) = (m' + x)^e - c = 0 \mod N$$

if $|m - m'| < N^{1/e}$.

Coppersmith's methods

In [Cop@EC'96a], [Cop@EC'96b], Coppersmith partially resolved the problem.

Coppersmith methods solve integer/modular equations with *small roots* in polynomial time.

So far, the methods reveal numerous cryptanalytic results especially for RSA and factorization problems.

The core trick is the LLL lattice basis reduction algorithm.

Next: Overview of the modular method due to [How@IMACC'97]

Coppersmith's method

Target modular equation

$$f(x) = 0 \mod N$$

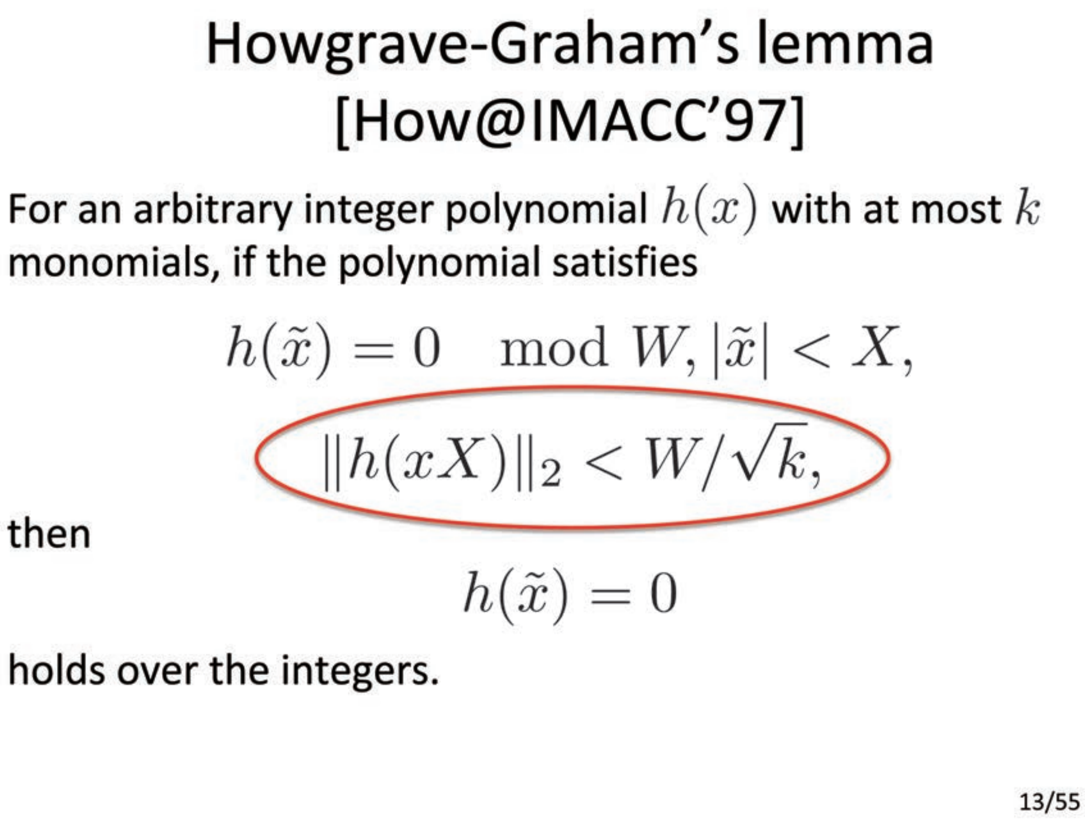
$f(x)$: monic, univariate polynomial
We want to find the root $\tilde{x}$ such that $|\tilde{x}| < X$.

If we can obtain a polynomial $g(x)$ such that

$$g(\tilde{x}) = 0$$

holds over the integers, then we can recover the root $\tilde{x}$.

How can we obtain $g(x)$?



Howgrave-Graham's lemma [How@IMACC'97]

For an arbitrary integer polynomial $h(x)$ with at most k monomials, if the polynomial satisfies

$$h(\tilde{x}) = 0 \mod W, |\tilde{x}| < X,$$

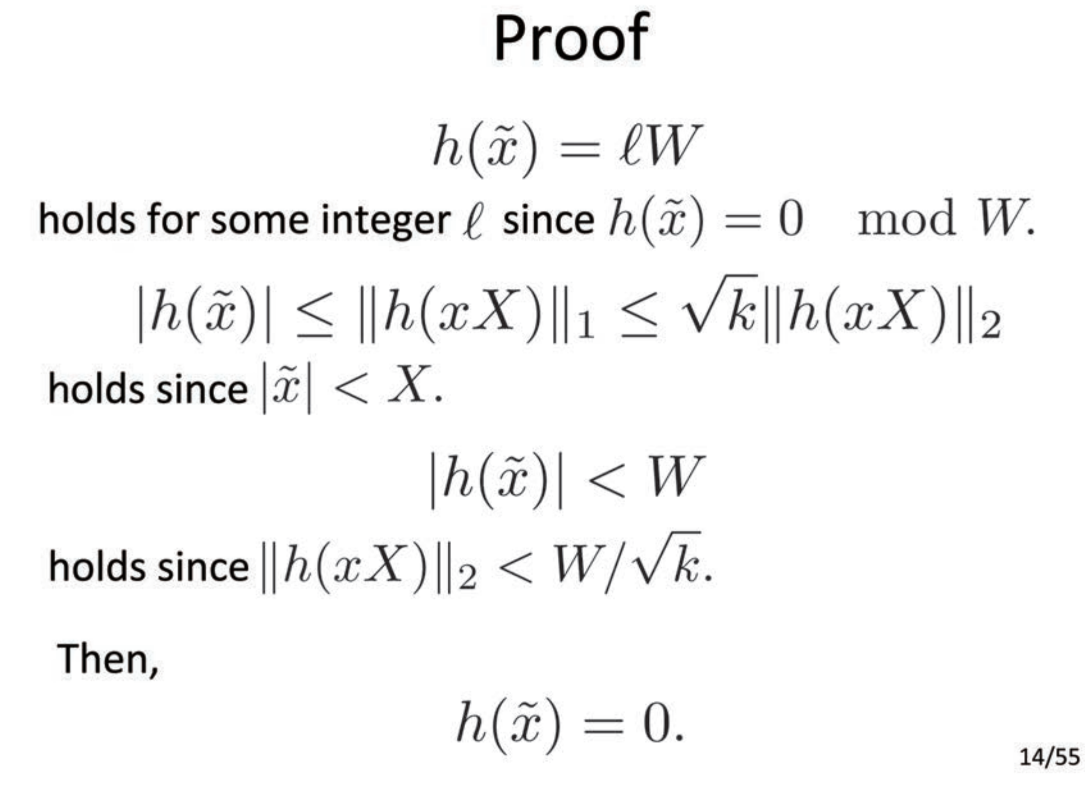
$$\boxed{\|h(xX)\|_2 < W/\sqrt{k},}$$

then

$$h(\tilde{x}) = 0$$

holds over the integers.

13/55



Proof

$$h(\tilde{x}) = \ell W$$

holds for some integer ℓ since $h(\tilde{x}) = 0 \mod W.$

$$|h(\tilde{x})| \leq \|h(xX)\|_1 \leq \sqrt{k}\|h(xX)\|_2$$

holds since $|\tilde{x}| < X.$

$$|h(\tilde{x})| < W$$

holds since $\|h(xX)\|_2 < W/\sqrt{k}.$

Then,

$$h(\tilde{x}) = 0.$$

14/55

Approach

To solve $f(x) = 0 \mod N$ and recover $\tilde{x}$,

Generate n polynomials $g_1(x), \ldots, g_n(x)$ such that

$$g_i(\tilde{x}) = 0 \mod N^m$$

for a positive integer m.

Then, we compute a *low norm* polynomial $h(x)$ by an *integer linear combination* of $g_1(x), \ldots, g_n(x)$.

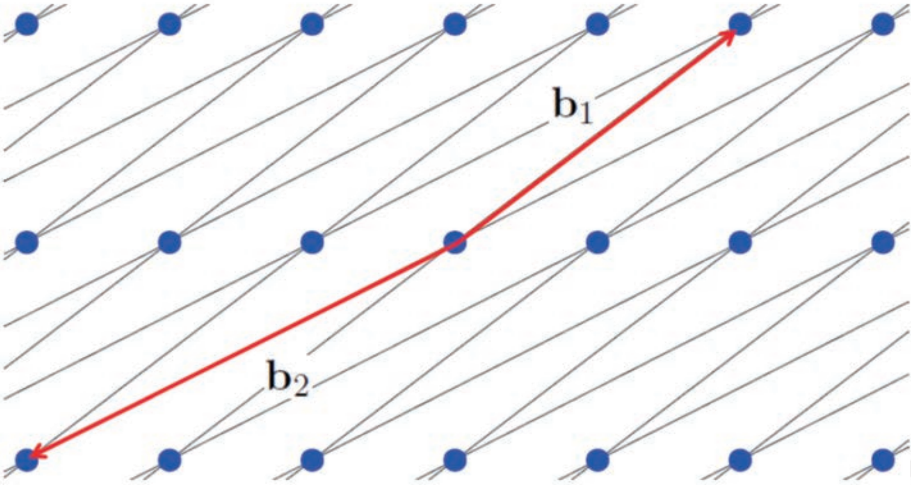
How can we do it?

⇨ LLL algorithm!

(Integer) Lattices

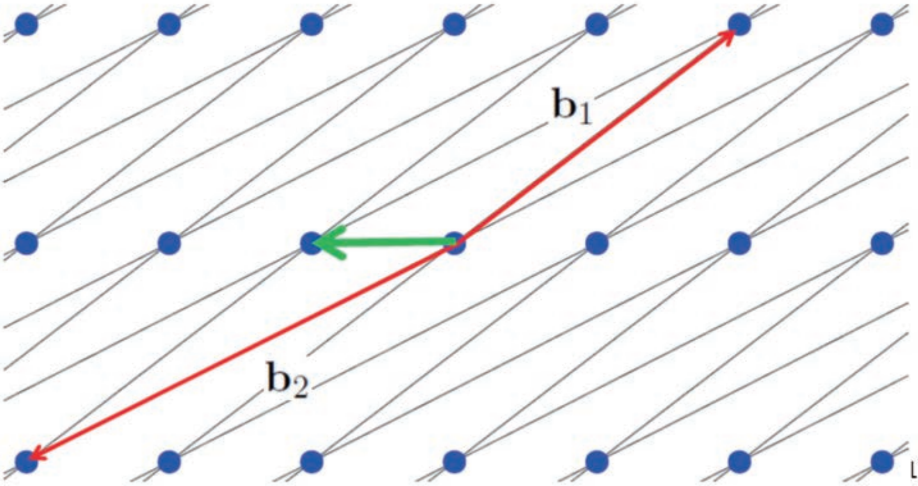
An additive discrete subgroup of $\mathbb{Z}^m$.

Integer linear combinations of a basis $\{\vec{b}_1, \ldots, \vec{b}_n\}$.



LLL algorithm

Given a basis $\{\vec{b}_1, \ldots, \vec{b}_n\}$, output another ``*<u>short</u>*'' basis $\{\vec{v}_1, \ldots, \vec{v}_n\}$ in polynomial time.



Output quality

Basis matrix: $\mathbf{B} := (\vec{b}_1, \ldots, \vec{b}_n)^T$

Volume of a lattice spanned by $\mathbf{B}$:

$$\begin{aligned} \mathrm{vol}(L(\mathbf{B})) &:= \sqrt{\det(\mathbf{B}\mathbf{B}^T)} \\ &= |\det(\mathbf{B})| \text{ for } n = m \end{aligned}$$

Norms of vectors output by LLL are roughly bounded by

$$2^{O(n)} \cdot \mathrm{vol}(L(\mathbf{B}))^{1/n}.$$

Approach

To solve $f(x) = 0 \mod N$ and recover $\tilde{x}$,

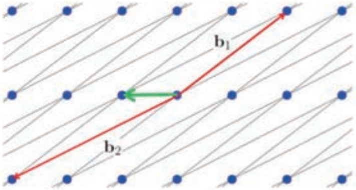
Generate n polynomials $g_1(x), \ldots, g_n(x)$ such that

$$g_i(\tilde{x}) = 0 \mod N^m$$

for a positive integer m.

Then, we compute a *low norm* polynomial $h(x)$ by an *integer linear combination* of $g_1(x), \ldots, g_n(x)$.

Utilizing LLL algorithm!



Approach using the LLL

To solve $f(x) = 0 \mod N$ and recover $\tilde{x}$,

Generate n polynomials $g_1(x), \ldots, g_n(x)$ such that

$$g_i(\tilde{x}) = 0 \mod N^m$$

for a positive integer m.

Let $\vec{b}_i$ be a coefficient vector of $g_i(xX)$.

Find a short vector $\vec{v}_1$ using the LLL reduction.

A polynomial $h(x)$ derived from $\vec{v}_1$ is an *integer linear combination* of $g_1(x), \ldots, g_n(x)$ and its *norm is small*.

Condition

$$h(\tilde{x}) = 0$$

holds over the integers when

$$\|h(xX)\|_2 < N^m/\sqrt{k}$$

$$\Updownarrow$$

$$2^{O(n)} \cdot \mathrm{vol}(L(\mathbf{B}))^{1/n} < N^m/\sqrt{k}$$

$$\wr\wr$$

$$|\det(\mathbf{B})|^{1/n} < N^m$$

Example for *m*=1

To solve $f(x) = a + bx + x^2 = 0 \mod N,$

Construct a matrix

$$\mathbf{B} = \begin{pmatrix} N & & \\ & NX & \\ a & bX & X^2 \end{pmatrix}$$

for $m = 1,$ with $g_1(x) = N$

$g_2(x) = Nx$

$g_3(x) = f(x)$

that have the same root modulo $N.$

Coefficients of each polynomial $g_i(xX)$

Example for *m*=1

To solve $f(x) = a + bx + x^2 = 0 \mod N,$

Construct a matrix

$$\mathbf{B} = \begin{pmatrix} N & & \\ & NX & \\ a & bX & X^2 \end{pmatrix}$$

Coefficients of each variable *x^j*

for $m = 1,$ with $g_1(x) = N$

$$g_2(x) = Nx$$

$$g_3(x) = f(x)$$

that have the same root modulo N.

Condition for *m*=1

To solve $f(x) = a + bx + x^2 = 0 \mod N,$

Construct a matrix

$$\mathbf{B} = \begin{pmatrix} N & & \\ & NX & \\ a & bX & X^2 \end{pmatrix}$$

Condition:

$$|\det(\mathbf{B})|^{1/n} = N^{2/3}X < N$$

$$\Rightarrow \quad X < N^{1/3}$$

Example for $m=2$

Coefficients of each polynomial $g_i(xX)$

To solve $f(x) = a + bx + x^2 = 0$

Construct a matrix

$$\mathbf{B} = \begin{pmatrix} N^2 & & & & \\ & N^2X & & & \\ Na & NbX & NX^2 & & \\ & NaX & NbX^2 & NX^3 & \\ a^2 & 2abX & (2a+b^2)X^2 & 2bX^3 & X^4 \end{pmatrix}$$

for $m = 2$, with $N^2, N^2x, Nf(x), Nxf(x), f^2(x)$ that have the same root modulo N^2.

Example for $m=2$

Coefficients of each variable x^j

To solve $f(x) = a + bx + x^2 = 0$

Construct a matrix

$$\mathbf{B} = \begin{pmatrix} N^2 & & & & \\ & N^2X & & & \\ Na & NbX & NX^2 & & \\ & NaX & NbX^2 & NX^3 & \\ a^2 & 2abX & (2a+b^2)X^2 & 2bX^3 & X^4 \end{pmatrix}$$

for $m = 2$, with $N^2, N^2x, Nf(x), Nxf(x), f^2(x)$ that have the same root modulo N^2.

Condition for *m*=2

To solve $f(x) = a + bx + x^2 = 0 \mod N,$

Construct a matrix

$$\mathbf{B} = \begin{pmatrix} N^2 & & & & \\ & N^2X & & & \\ Na & NbX & NX^2 & & \\ & NaX & NbX^2 & NX^3 & \\ a^2 & 2abX & (2a+b^2)X^2 & 2bX^3 & X^4 \end{pmatrix}$$

Condition:

$$|\det(\mathbf{B})|^{1/n} = N^{6/5}X^{10/5} < N^2$$

$$\Rightarrow \quad X < N^{2/5}$$

Example for general *m*

Coefficients of each polynomial *g*$_i$(*xX*)

To solve $f(x) = a + bx + x^2 = 0$

Construct a matrix

$$\mathbf{B} = \begin{pmatrix} N^m & & & & \\ & N^mX & & & \\ N^{m-1}a & N^{m-1}bX & N^{m-1}X^2 & & \\ \vdots & & & \ddots & \\ \cdots & & & \cdots & X^m \end{pmatrix}$$

with $(N^{m-i}f^i, N^{m-i}xf^i)_{i=0,1,\ldots,m-1}, f^m(x)$ that have the same root modulo N^m.

Example for general *m*

Coefficients of each variable x^j

To solve $f(x) = a + bx + x^2 = 0$

Construct a matrix

$$\mathbf{B} = \begin{pmatrix} N^m & & & & \\ & N^m X & & & \\ N^{m-1}a & N^{m-1}bX & N^{m-1}X^2 & & \\ \vdots & & & \ddots & \\ \cdots & & & \cdots & X^m \end{pmatrix}$$

with $(N^{m-i}f^i, N^{m-i}xf^i)_{i=0,1,\dots,m-1}, f^m(x)$ that have the same root modulo N^m.

Condition for general *m*

To solve $f(x) = a + bx + x^2 = 0 \mod N,$

Construct a matrix

$$\mathbf{B} = \begin{pmatrix} N^m & & & & \\ & N^m X & & & \\ N^{m-1}a & N^{m-1}bX & N^{m-1}X^2 & & \\ \vdots & & & \ddots & \\ \cdots & & & \cdots & X^m \end{pmatrix}$$

Condition:

$$|\det(\mathbf{B})|^{1/n} = N^{m(m+1)/(2m+1)}X^m < N^m$$

$\Rightarrow$ $X < N^{1/2}$ for large m

Multivariate extension

To solve $f(x_1, \ldots, x_k) = 0 \mod N$ and recover $(\tilde{x}_1, \ldots, \tilde{x}_k)$ s.t. $|\tilde{x}_j| < X_j$,

Generate n polynomials $(g_i(x_1, \ldots, x_k))_{i=1,2,\ldots,n}$ s.t.

$$g_i(\tilde{x}_1, \ldots, \tilde{x}_k) = 0 \mod N^m$$

for a positive integer m.

Let $\vec{b}_i$ be a coefficient vector of $g_i(x_1X_1, \ldots, x_kX_k)$.

Find short vectors $\vec{v}_1, \ldots, \vec{v}_k$ using the LLL reduction.

A polynomial $h_i(x_1, \ldots, x_k)$ derived from $\vec{v}_i$ is an integer linear combination of $g_i(x_1, \ldots, x_k)$ and its norm is small.

Multivariate Coppersmith Heuristic

Can we also solve the equation if

$$|\det(\mathbf{B})|^{1/n} < N^m?$$

Unfortunately, although $\vec{v}_i$'s are linearly independent vectors, there are no assurance that $h_i(x_1, \ldots, x_k)$'s are algebraically independent.

Since $h_i(x_1, \ldots, x_k)$'s are algebraically independent in practice, we assume the fact in multivariate cases.

Research direction

- Towards resolving the multivariate heuristic [BJ@EC'07]
- Proof of the optimality [AASW@ACISP'12],[CHHS@AC'16]
- Utilizing existing algorithms[BCC+@AC'13],[NSS+@CCS'17]
- Speed-up the implementation [BCF+@PKC'14]
- Constructing new (multivariate) algorithms
  - General lattice construction strategy [JM@AC'06],[TK@ACISP'13]
  - Small secret exponent attack [BD@IEEE TIT'00], [HM@PKC'10]
  - Partial key exposure attack [BM@Crypto'03],[EJMW@EC'05], [TK@SAC'14],[TK@CT-RSA'17]
  - Small CRT exponent attack [TLP@EC'17]
  - Attacks on RSA variants [LZPL@AC'15],[TK@PKC'15]

Small CRT-exponent attack

A. Takayasu, Y. Lu, and L. Peng. Small CRT-exponent RSA revisited.
Proc. Eurocrypt 2017.
Journal of Cryptology 2018.
IACR ePrint: 2017/092

Small secret exponent attack

Public Key: $(N = pq, e)$ Secret Key: (p, q, d)

$$ed = 1 \mod (p-1)(q-1)$$

[Boneh-Durfee@IEEE TIT'00] proposed a polynomial time factorization attack on RSA for

$$d < N^{0.292}.$$

35/55

CRT-RSA

Public Key: $(N = pq, e)$ Secret Key: (p, q, d_p, d_q)

$$ed_p = 1 \mod p-1, \quad ed_q = 1 \mod q-1$$

Are there analogous polynomial time factorization attacks for small CRT-exponents?

36/55

Small d_q attack

- p is significantly smaller than q.
  d_q is significantly smaller than q.

Can we reach $p<N^{0.5}$?

- [May@Crypto'02]

$$p < N^{0.382}$$

- [Bleichenbacher-May@PKC'06]

$$p < N^{0.468}$$

✓ Variants: [SIK@IEICE Trans.'11],[PHL+@Indocrypt'15]

✓ Extensions: [BM@Crypto'03],[LZL@ACNS'14], [TK@ACNS'15],[TK@ISC'16]

Small d_p and d_q attack

- p, q are the same bit-size.
  d_p, d_q are significantly smaller than p, q.
- [Jochemsz-May@Crypto'07]

$$d_p, d_q < N^{0.073}$$

✓ Extensions: [SM@ACNS'09],[TK@ACNS'15]

Can we improve the bound?

Our results

- Small d_q attack

$$p < N^{0.5}$$

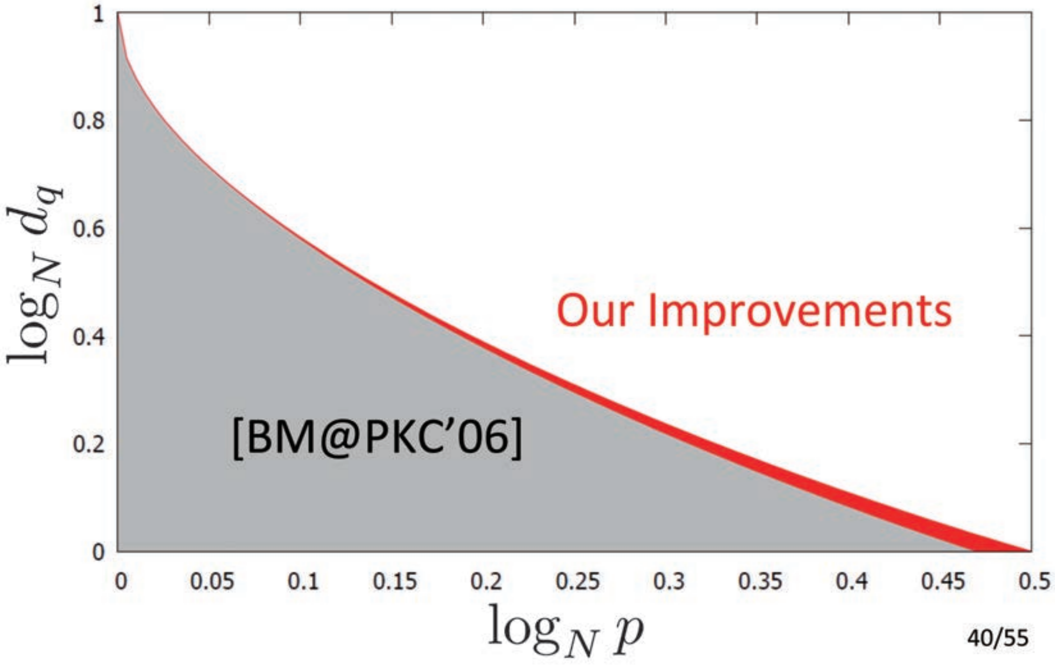
- Small d_p and d_q attack

$$d_p, d_q < N^{0.122}$$

- Improved attacks on the variants

✓ Improved lattice constructions that are specialized to CRT-RSA key generation.

Comparison



Formulation

Solving the latter equation seems better approach since p is much smaller than q.

There is an integer k suc

- We can factorize N b

$$f_q(x_q, y_q) = 1 + x_q(y_q - 1) = 0 \mod e$$

and recover $(x_q, y_q) = (k, q)$.

- Multiplying p :

$$ed_q p = p + k(N - p) = N + (k-1)(N - p)$$

We can factorize N by solving a modular equation

$$f_p(x_p, y_p) = N + x_p(N - y_p) = 0 \mod e$$

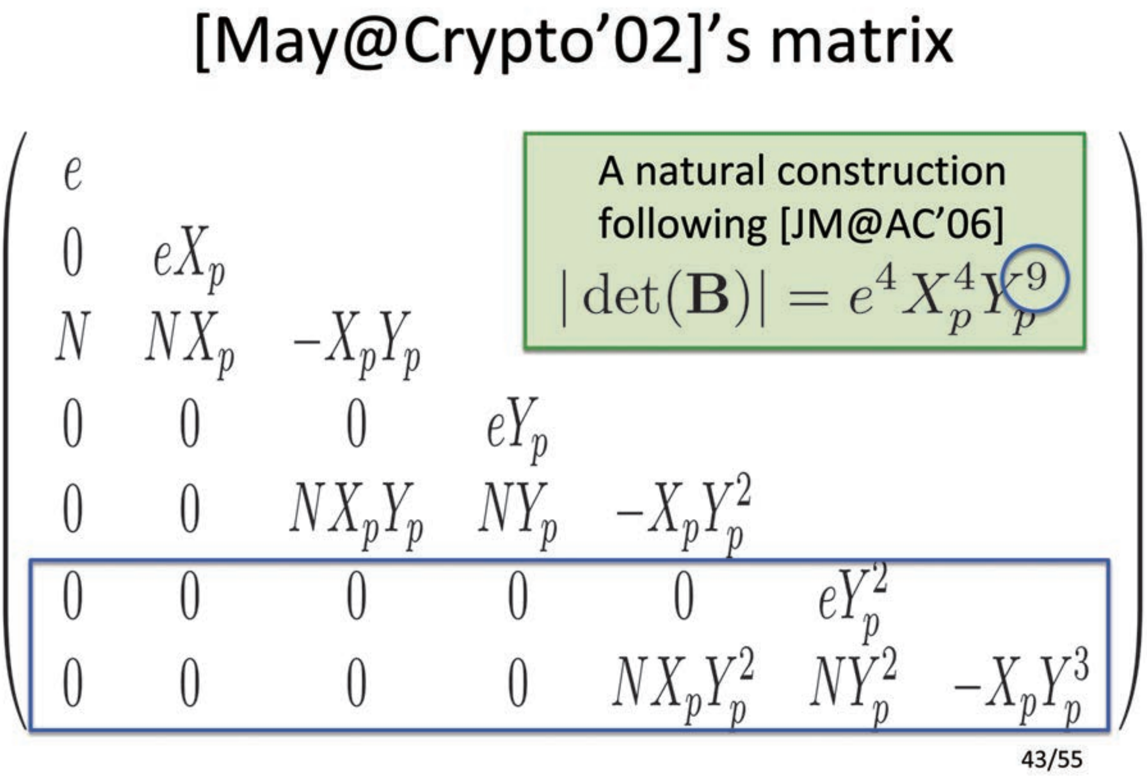
and recover $(x_p, y_p) = (k-1, p)$.

[May@Crypto'02]'s matrix

$$f_p(x_p, y_p) = N + x_p(N - y_p)$$

$$y_p^2 f_p(x_p, y_p)$$

$$\begin{pmatrix}
e & & & & & & \\
0 & eX_p & & & & & \\
N & NX_p & -X_pY_p & & & & \\
0 & 0 & 0 & eY_p & & & \\
0 & 0 & NX_pY_p & NY_p & -X_pY_p^2 & & \\
0 & 0 & 0 & 0 & 0 & eY_p^2 & \\
0 & 0 & 0 & 0 & NX_pY_p^2 & NY_p^2 & -X_pY_p^3
\end{pmatrix}$$



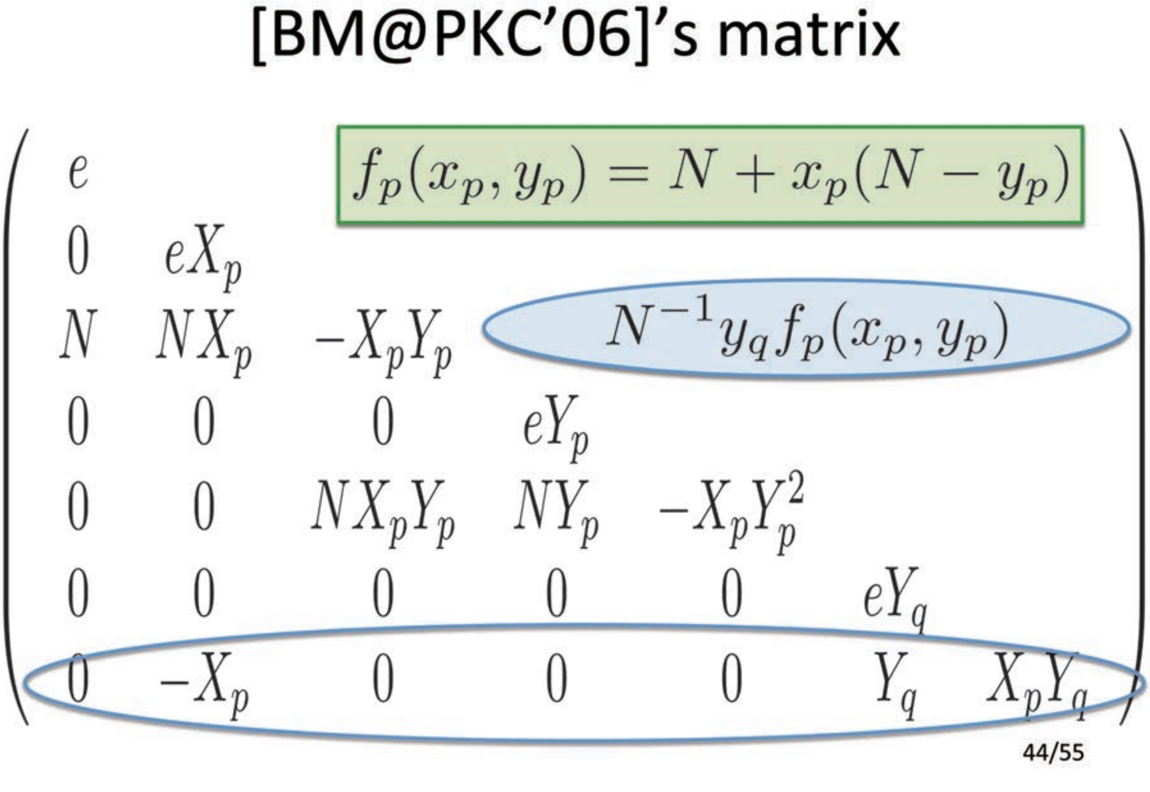
[May@Crypto'02]'s matrix

$$\begin{pmatrix} e & & & & & & \\ 0 & eX_p & & & & & \\ N & NX_p & -X_pY_p & & & & \\ 0 & 0 & 0 & eY_p & & & \\ 0 & 0 & NX_pY_p & NY_p & -X_pY_p^2 & & \\ 0 & 0 & 0 & 0 & 0 & eY_p^2 & \\ 0 & 0 & 0 & 0 & NX_pY_p^2 & NY_p^2 & -X_pY_p^3 \end{pmatrix}$$

A natural construction following [JM@AC'06]

$$|\det(\mathbf{B})| = e^4X_p^4Y_p^9$$

43/55



[BM@PKC'06]'s matrix

$$f_p(x_p, y_p) = N + x_p(N - y_p)$$

$$\begin{pmatrix} e & & & & & & \\ 0 & eX_p & & & & & \\ N & NX_p & -X_pY_p & & & & \\ 0 & 0 & 0 & eY_p & & & \\ 0 & 0 & NX_pY_p & NY_p & -X_pY_p^2 & & \\ 0 & 0 & 0 & 0 & 0 & eY_q & \\ 0 & -X_p & 0 & 0 & 0 & Y_q & X_pY_q \end{pmatrix}$$

$$N^{-1}y_qf_p(x_p, y_p)$$

44/55

[BM@PKC'06]'s matrix

$$\begin{pmatrix} e & & & & & & \\ 0 & eX_p & & & & & \\ N & NX_p & -X_pY_p & & & & \\ 0 & 0 & 0 & eY_p & & & \\ 0 & 0 & NX_pY_p & NY_p & -X_pY_p^2 & & \\ 0 & 0 & 0 & 0 & 0 & eY_q & \\ 0 & -X_p & 0 & 0 & 0 & Y_q & X_pY_q \end{pmatrix}$$

Reducing the powers of Y_p

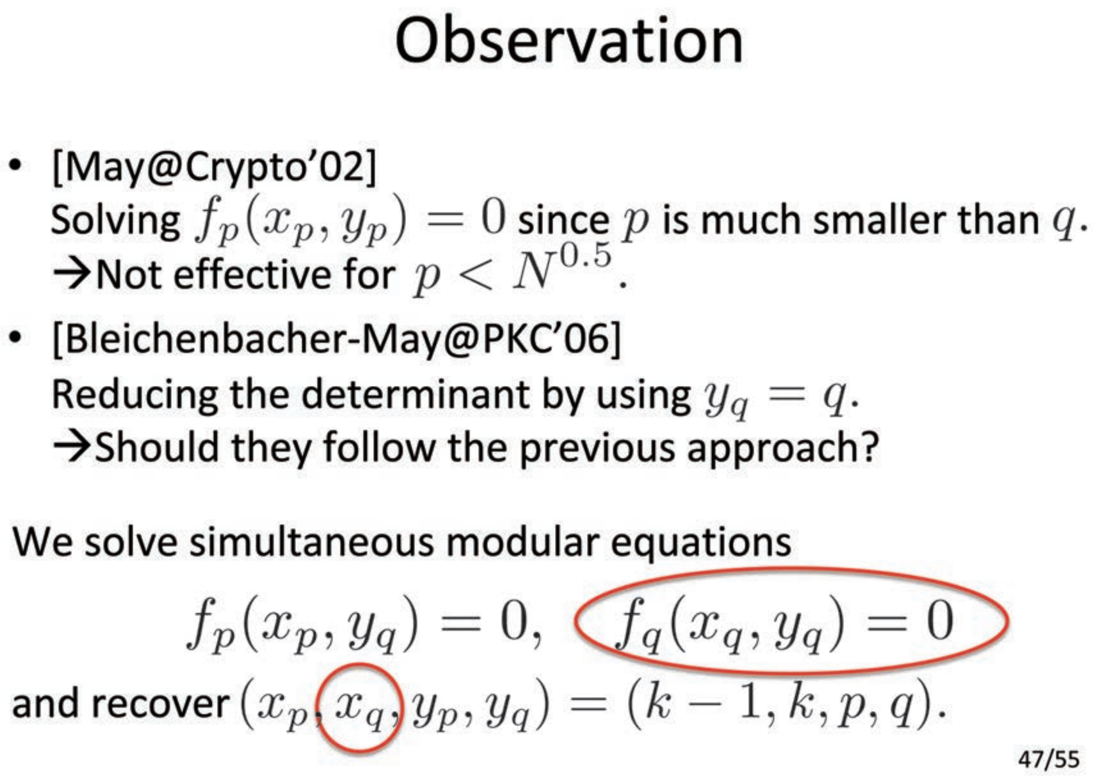
$$|\det(\mathbf{B})| = e^4X_p^4Y_p^9$$

$$\to |\det(\mathbf{B})| = e^4X_p^4Y_p^4Y_q^2$$

[BM@PKC'06]'s matrix

$$\begin{pmatrix} e & & & & & & \\ 0 & eX_p & & & & & \\ N & NX_p & -X_pY_p & & & & \\ 0 & 0 & 0 & eY_p & & & \\ 0 & 0 & NX_pY_p & NY_p & -X_pY_p^2 & & \\ 0 & 0 & 0 & 0 & 0 & eY_q & \\ 0 & -X_p & 0 & 0 & 0 & Y_q & X_pY_q \end{pmatrix}$$

By using $y_py_q = N$, either y_p or y_q appears in each monomial.



Observation

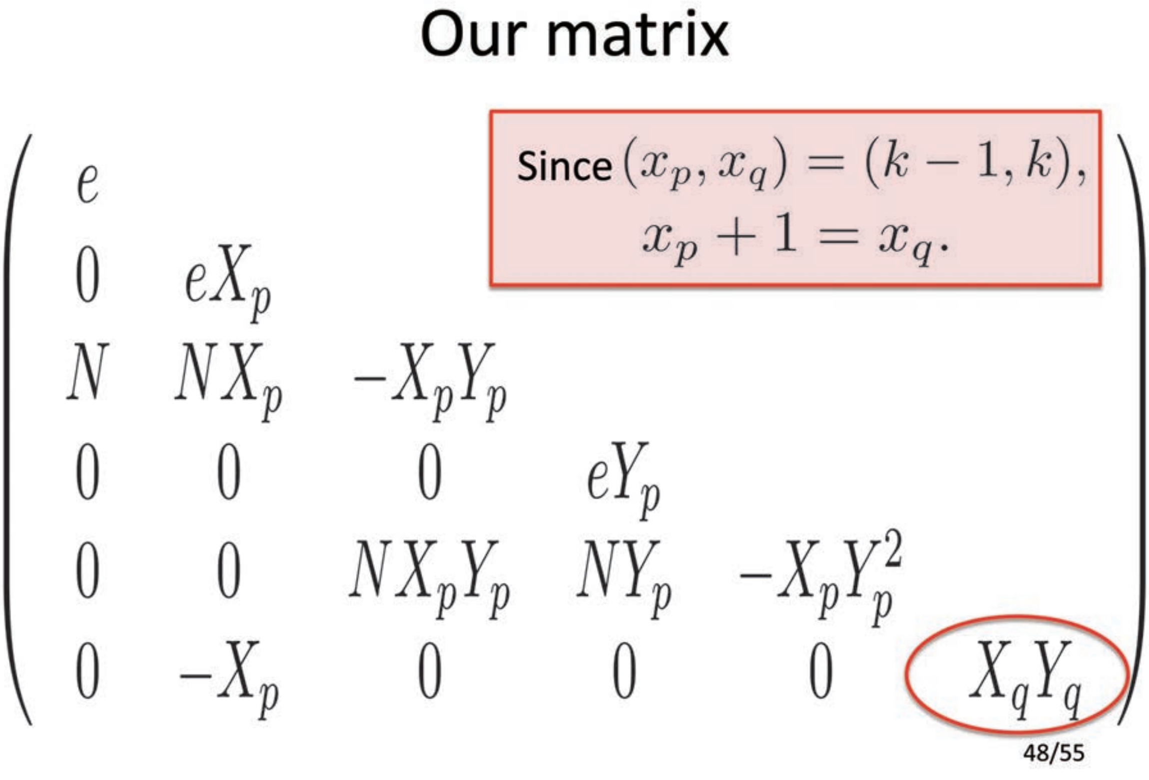
- [May@Crypto'02]
  Solving $f_p(x_p, y_p) = 0$ since p is much smaller than q.
  →Not effective for $p < N^{0.5}$.
- [Bleichenbacher-May@PKC'06]
  Reducing the determinant by using $y_q = q$.
  →Should they follow the previous approach?

We solve simultaneous modular equations

$$f_p(x_p, y_q) = 0, \quad f_q(x_q, y_q) = 0$$

and recover $(x_p, x_q, y_p, y_q) = (k-1, k, p, q)$.

47/55

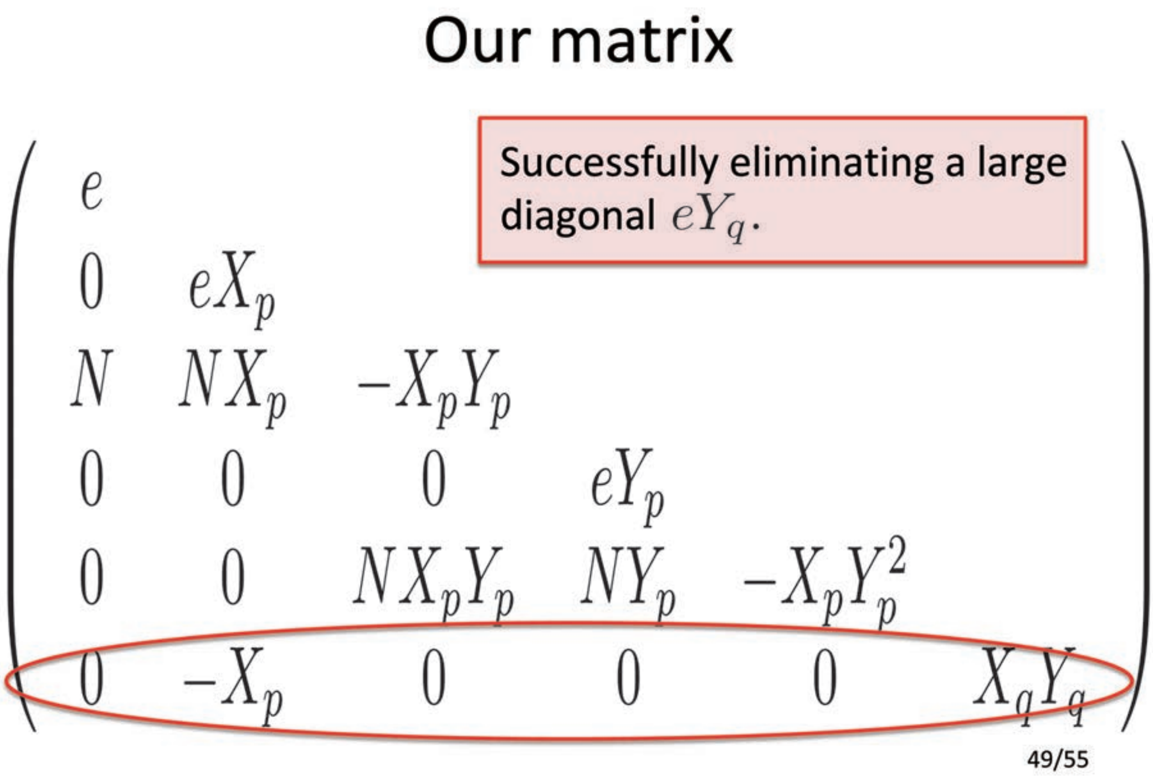


Our matrix

Since $(x_p, x_q) = (k-1, k)$,
$x_p + 1 = x_q$.

$$\begin{pmatrix} e & & & & & \\ 0 & eX_p & & & & \\ N & NX_p & -X_pY_p & & & \\ 0 & 0 & 0 & eY_p & & \\ 0 & 0 & NX_pY_p & NY_p & -X_pY_p^2 & \\ 0 & -X_p & 0 & 0 & 0 & X_qY_q \end{pmatrix}$$

48/55

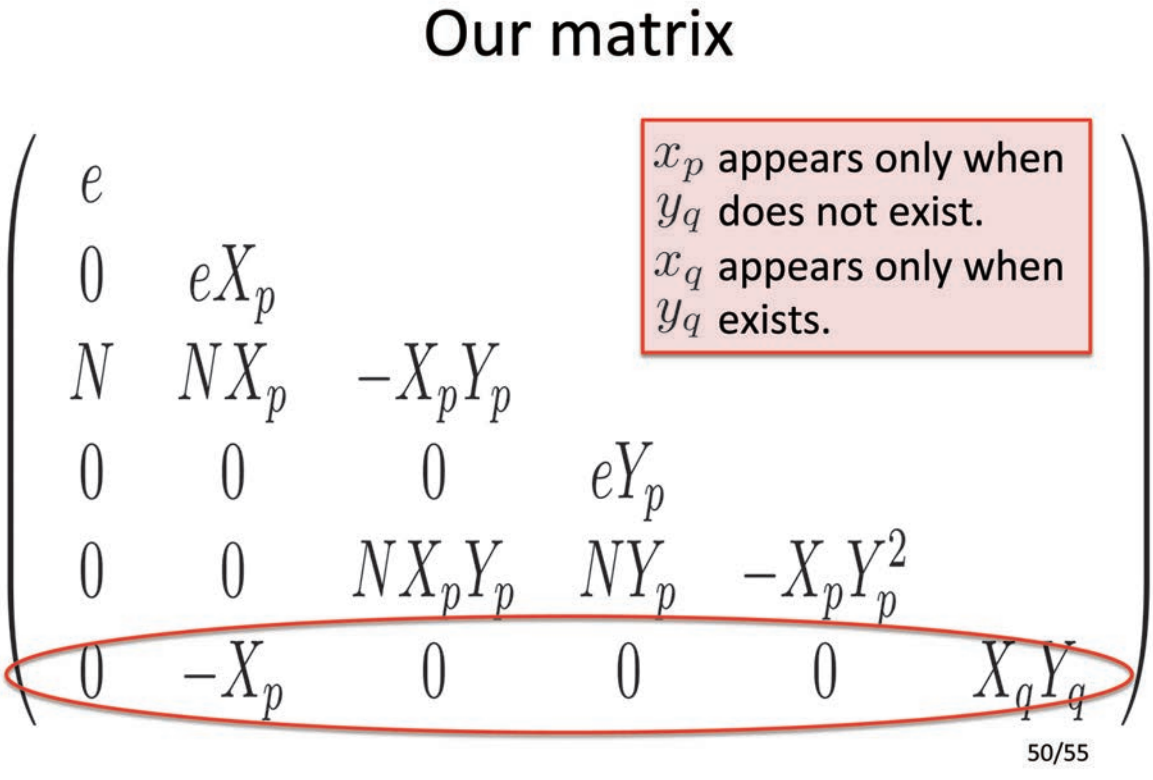


Our matrix

Successfully eliminating a large diagonal eY_q.

$$\begin{pmatrix} e & & & & & \\ 0 & eX_p & & & & \\ N & NX_p & -X_pY_p & & & \\ 0 & 0 & 0 & eY_p & & \\ 0 & 0 & NX_pY_p & NY_p & -X_pY_p^2 & \\ 0 & -X_p & 0 & 0 & 0 & X_qY_q \end{pmatrix}$$

49/55



Our matrix

x_p appears only when y_q does not exist. x_q appears only when y_q exists.

$$\begin{pmatrix} e & & & & & \\ 0 & eX_p & & & & \\ N & NX_p & -X_pY_p & & & \\ 0 & 0 & 0 & eY_p & & \\ 0 & 0 & NX_pY_p & NY_p & -X_pY_p^2 & \\ 0 & -X_p & 0 & 0 & 0 & X_qY_q \end{pmatrix}$$

50/55

Our matrix

$\boxed{m = 1, \lambda = 1/2}$

$$\begin{pmatrix} e & & & & & \\ 0 & eX_p & & & & \\ N & NX_p & -X_pY_p & & & \\ 0 & 0 & 0 & eY_p & & \\ 0 & 0 & NX_pY_p & NY_p & -X_pY_p^2 & \\ 0 & -X_p & 0 & 0 & 0 & X_qY_q \end{pmatrix}$$

Our matrix

$\boxed{m = 3, \lambda = 1/2}$

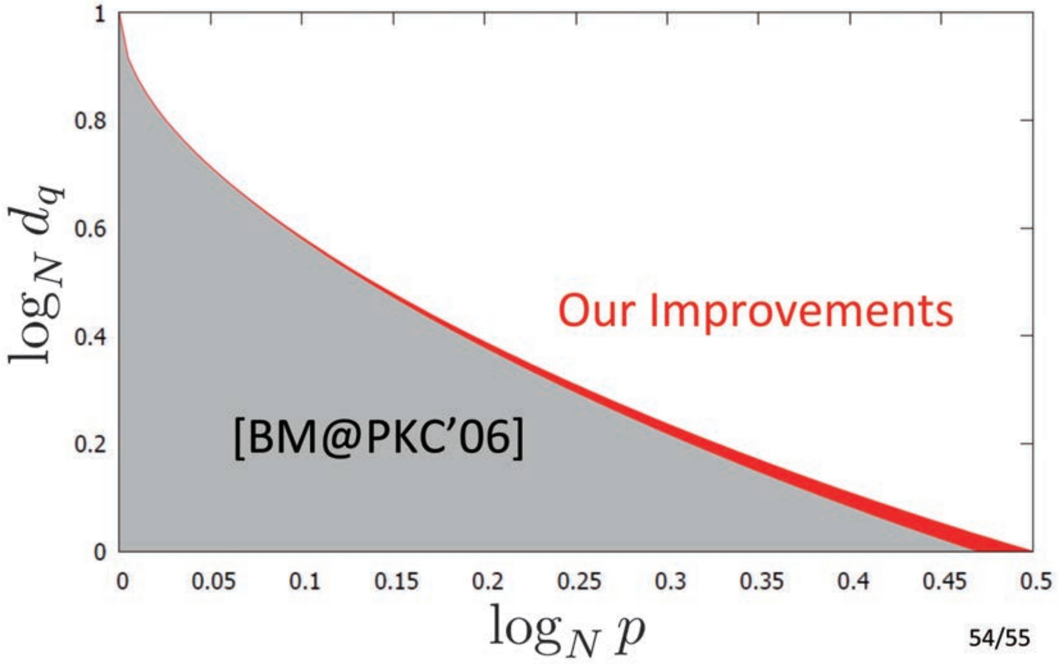
$$\begin{pmatrix} e^3 & & & & & & & & & & & \\ & X_pe^3 & & & & & & & & & & \\ & & X_p^2e^3 & & & & & & & & & \\ & & & X_p^3e^3 & & & & & & & & \\ Ne^2 & NX_pe^2 & & & -X_pY_pe^2 & & & & & & & \\ & NX_pe^2 & NX_p^2e^2 & & & -X_p^2Y_pe^2 & & & & & & \\ & & NX_p^2e^2 & NX_p^3e^2 & & & -X_p^3Y_pe^2 & & & & & \\ & -2X_pe & -2X_p^2e & & & N^{-1}X_p^2Y_pe & & X_q^2Y_qe & & & & \\ & & -2X_p^2e & -2X_p^3e & & & N^{-1}X_p^3Y_pe & -X_q^2Y_qe & X_q^3Y_qe & & & \\ & -3N^2X_p & -6N^2X_p^2 & -3N^2X_p^3 & & 3NX_p^2Y_p & 3NX_p^3Y_p & & N^2X_q^3Y_q & -X_p^3Y_p^2 & & \\ & -X_pe^2 & & & & & & & & & X_qY_qe^2 & \\ & & 3X_p^2 & 3X_p^3 & & & -3N^{-1}X_p^3Y_p & 3X_q^2Y_q & -3X_q^3Y_q & & & X_q^3Y_q^2 \end{pmatrix}$$

Our matrix

$$\boxed{m = 3, \lambda = 2/3}$$

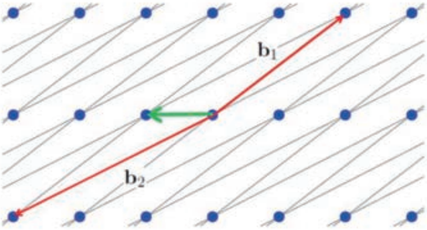
$$\begin{pmatrix}
e^3 &&&&&&&&&&&&& \\
& X_pe^3 &&&&&&&&&&&& \\
&& X_p^2e^3 &&&&&&&&&&& \\
&&& X_p^3e^3 &&&&&&&&&& \\
Ne^2 & NX_pe^2 &&& -X_pY_pe^2 &&&&&&&&& \\
& NX_pe^2 & NX_p^2e^2 &&& -X_p^2Y_pe^2 &&&&&&&& \\
&& NX_p^2e^2 & NX_p^3e^2 &&& -X_p^3Y_pe^2 &&&&&&& \\
N^2e & 2N^2X_pe & N^2X_p^2e && -2NX_pY_pe & -2NX_p^2Y_pe && X_p^2Y_p^2e &&&&&& \\
& N^2X_pe & 2N^2X_p^2e & N^2X_p^3e && -2NX_p^2Y_pe & -2NX_p^3Y_pe && X_p^3Y_p^2e &&&&& \\
& -3X_p & -6X_p^2 & -3X_p^3 && 3N^{-1}X_p^2Y_p & 3N^{-1}X_p^3Y_p && -N^{-2}X_p^3Y_p^2 & X_q^3Y_q &&&& \\
& -X_pe^2 &&&&&&&&& X_qY_qe^2 &&& \\
& -2X_pe & -2X_p^2e &&& N^{-1}X_p^2Y_pe &&&&&& X_q^2Y_qe && \\
&& X_p^2e &&&&&&&& 2X_qY_qe & -2X_q^2Y_qe & X_q^2Y_q^2e & \\
&& 3X_p^2 & 3X_p^3 &&& -N^{-1}X_p^3Y_p &&& -3X_q^3Y_q && 3X_q^2Y_q && X_q^3Y_q^2
\end{pmatrix}$$

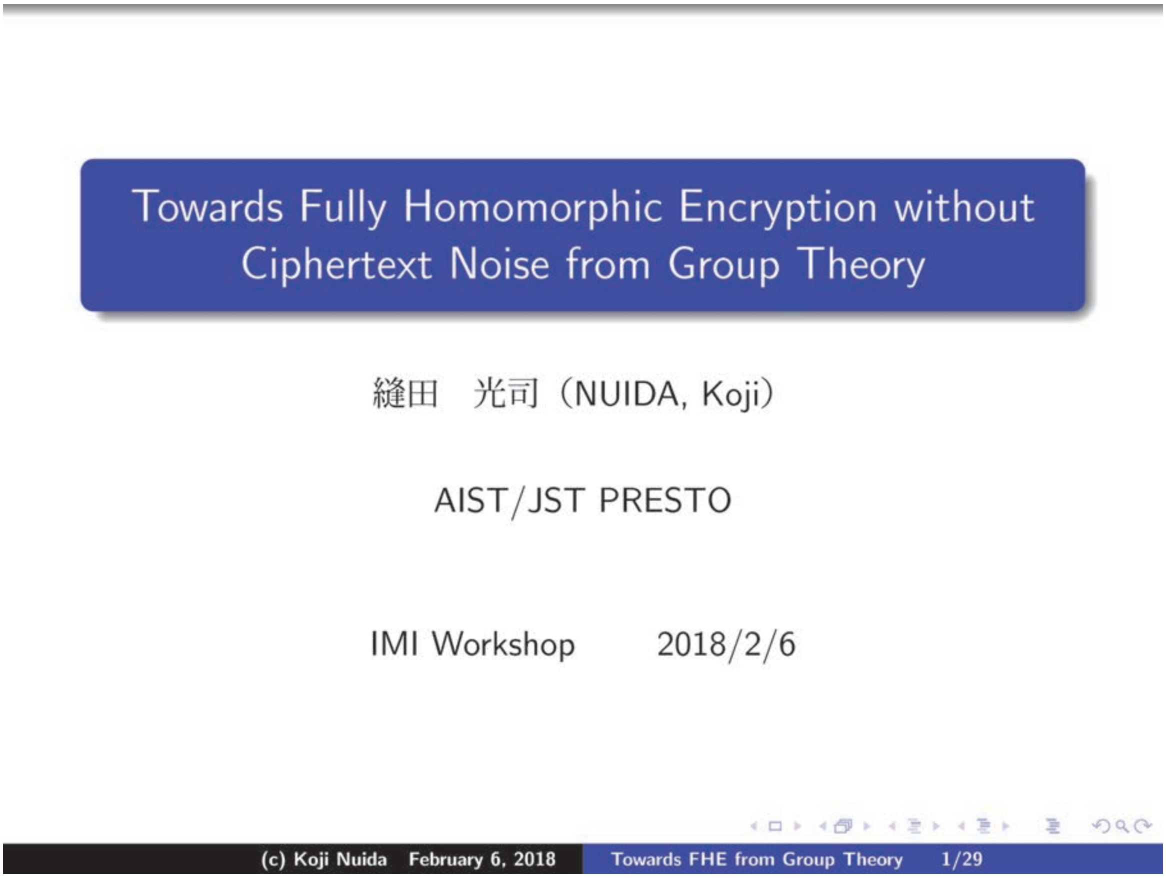
Comparison



Summary

- Coppersmith' methods
  Effective in the context of RSA and factorization problems.
  Polynomial time algorithms by using the LLL reduction.
- To obtain the better algorithms, we should work out constructing the better basis matrices.
- Introducing our small CRT-exponent attack [TLP@EC'17].



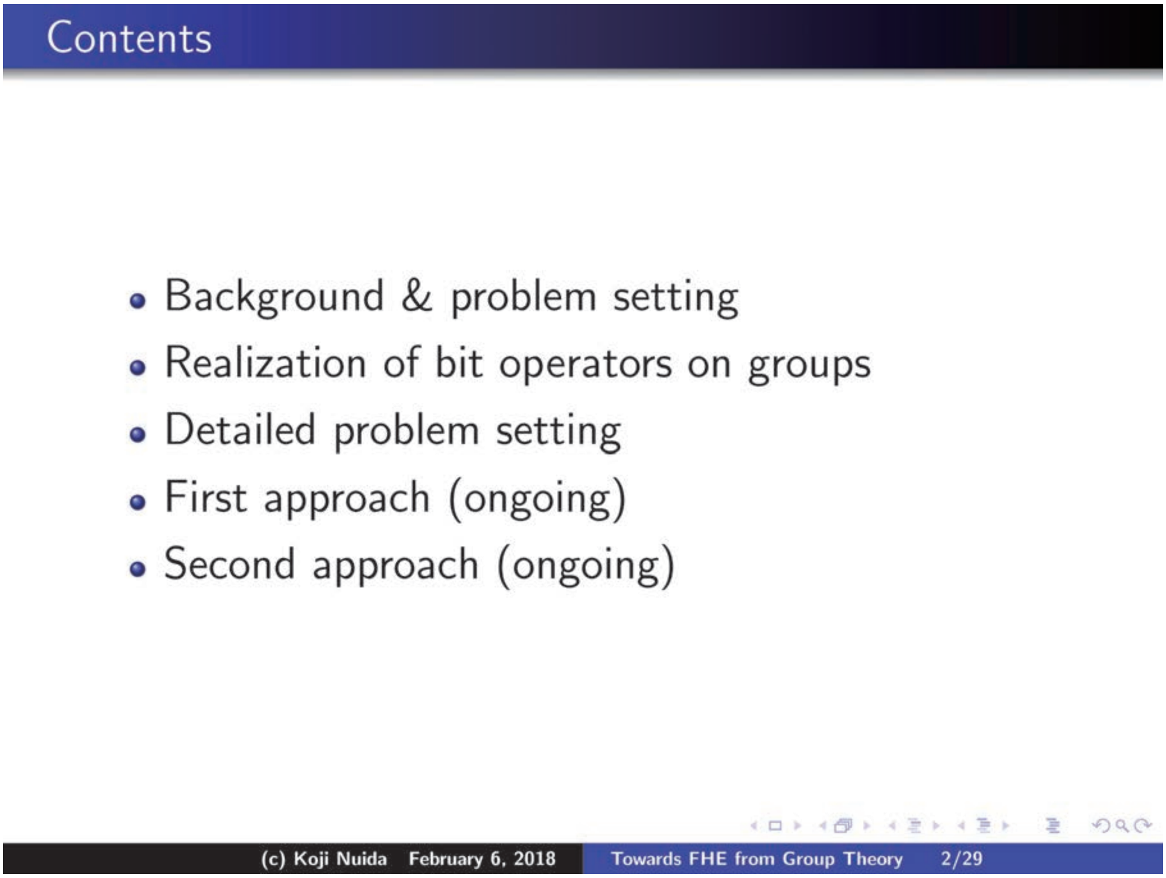


Towards Fully Homomorphic Encryption without Ciphertext Noise from Group Theory

縫田　光司（NUIDA, Koji）

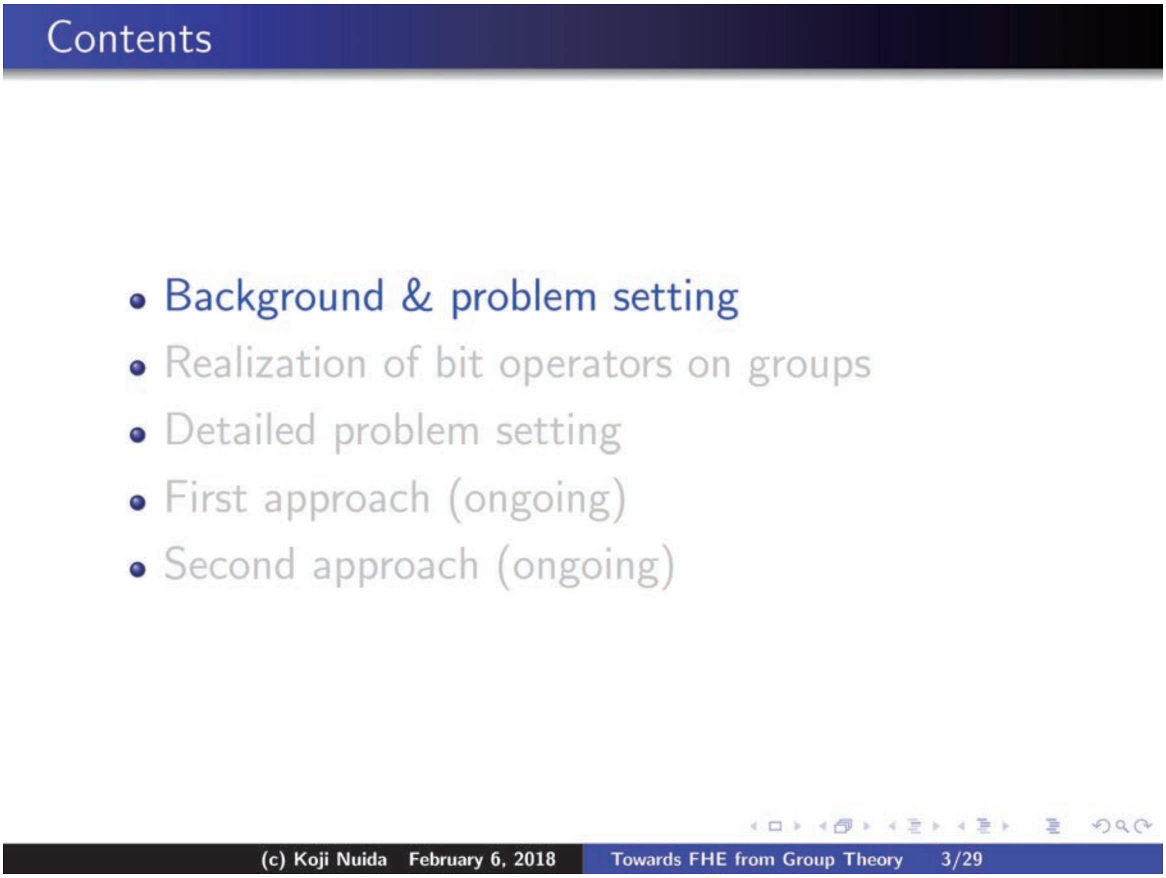
AIST/JST PRESTO

IMI Workshop　　2018/2/6



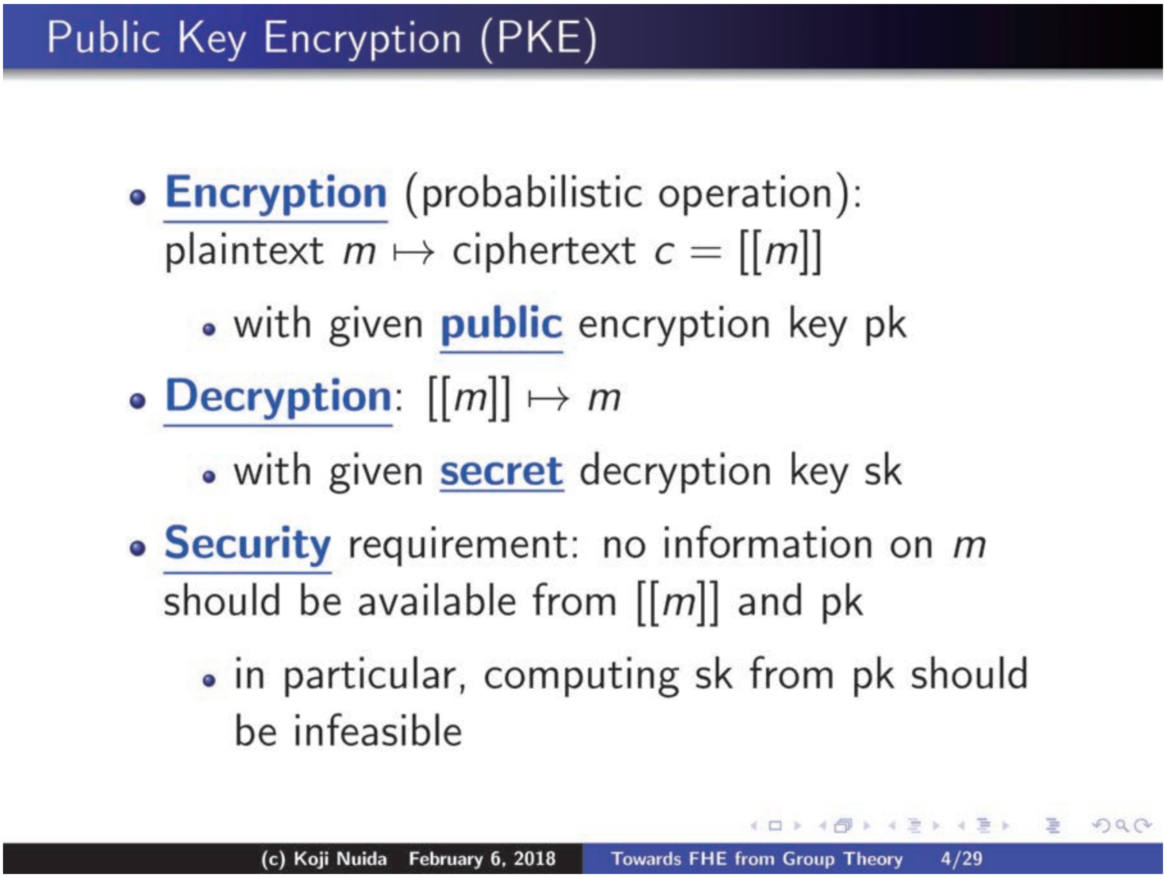
Contents

- Background & problem setting
- Realization of bit operators on groups
- Detailed problem setting
- First approach (ongoing)
- Second approach (ongoing)



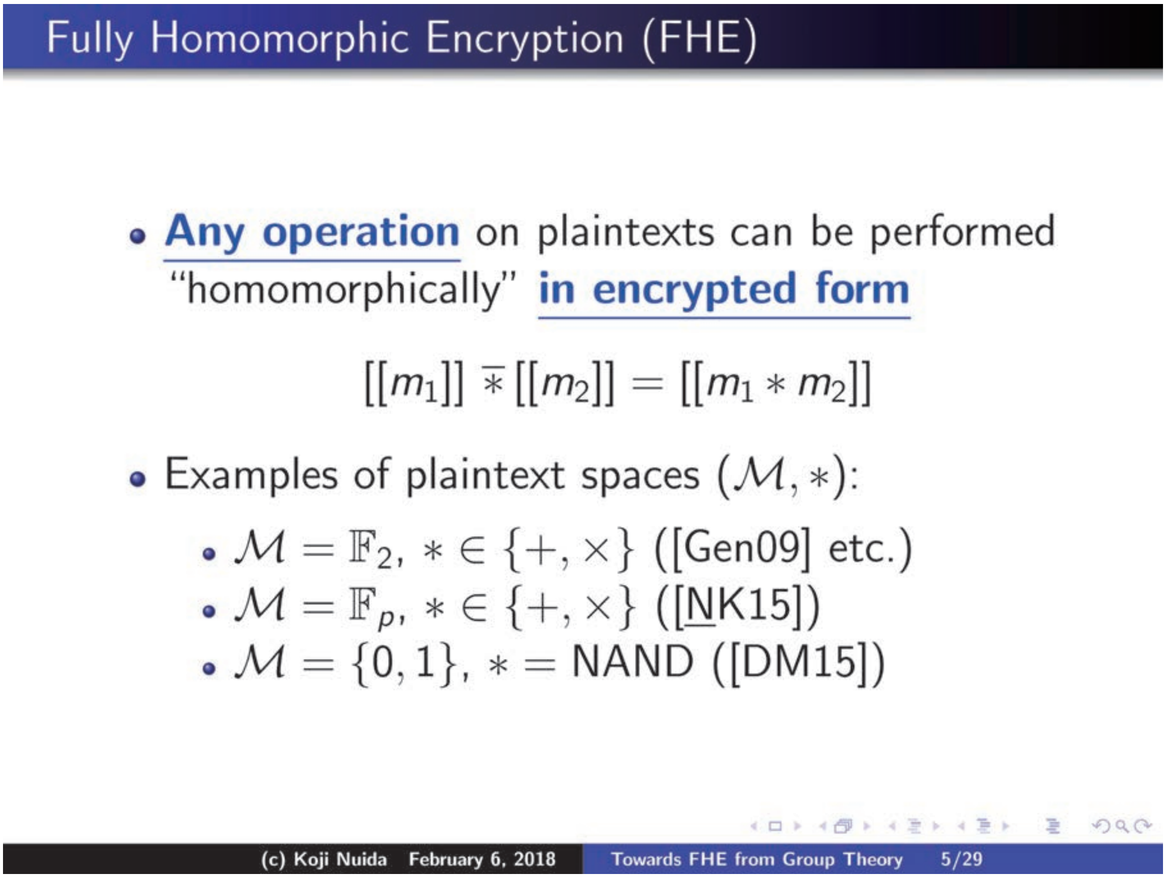
Contents

- Background & problem setting
- Realization of bit operators on groups
- Detailed problem setting
- First approach (ongoing)
- Second approach (ongoing)



Public Key Encryption (PKE)

- **Encryption** (probabilistic operation): plaintext $m \mapsto$ ciphertext $c = [[m]]$
  - with given **public** encryption key pk
- **Decryption**: $[[m]] \mapsto m$
  - with given **secret** decryption key sk
- **Security** requirement: no information on m should be available from $[[m]]$ and pk
  - in particular, computing sk from pk should be infeasible

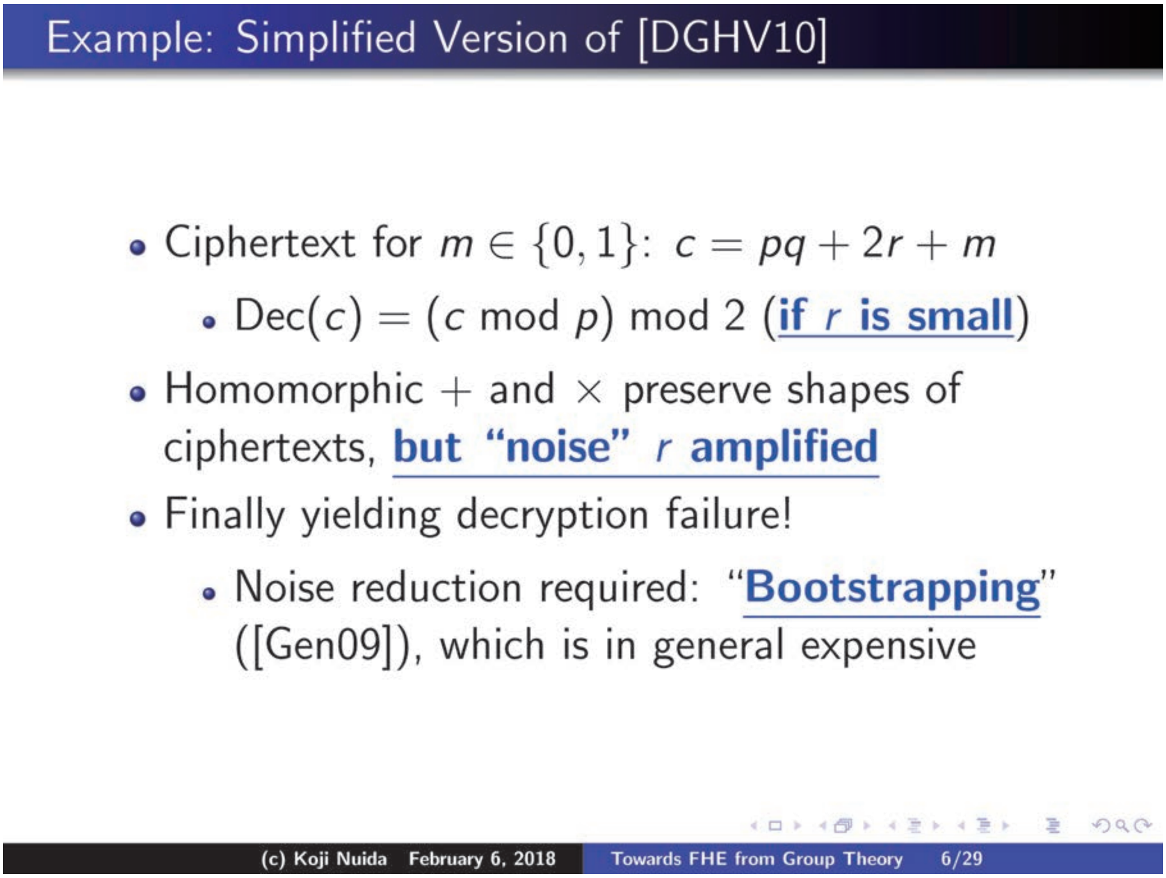


Fully Homomorphic Encryption (FHE)

- **Any operation** on plaintexts can be performed "homomorphically" **in encrypted form**

$$[[m_1]] \mathbin{\overline{*}} [[m_2]] = [[m_1 * m_2]]$$

- Examples of plaintext spaces $(\mathcal{M}, *)$:
  - $\mathcal{M} = \mathbb{F}_2$, $* \in \{+, \times\}$ ([Gen09] etc.)
  - $\mathcal{M} = \mathbb{F}_p$, $* \in \{+, \times\}$ ([NK15])
  - $\mathcal{M} = \{0, 1\}$, $* =$ NAND ([DM15])



Example: Simplified Version of [DGHV10]

- Ciphertext for $m \in \{0, 1\}$: $c = pq + 2r + m$
  - $\mathrm{Dec}(c) = (c \bmod p) \bmod 2$ (**if r is small**)
- Homomorphic $+$ and $\times$ preserve shapes of ciphertexts, **but "noise" r amplified**
- Finally yielding decryption failure!
  - Noise reduction required: "**Bootstrapping**" ([Gen09]), which is in general expensive

Existing Approaches

- High-dimensional lattices ([Gen09] etc.)
- "Almost nested" integer residue rings ([DGHV10] etc.)
- (Linear codes, but not succeeded)
- **My research: new approach to FHE**
  - towards "noise-free" FHE

Suggestion by Ostrovsky–Skeith III (2008)

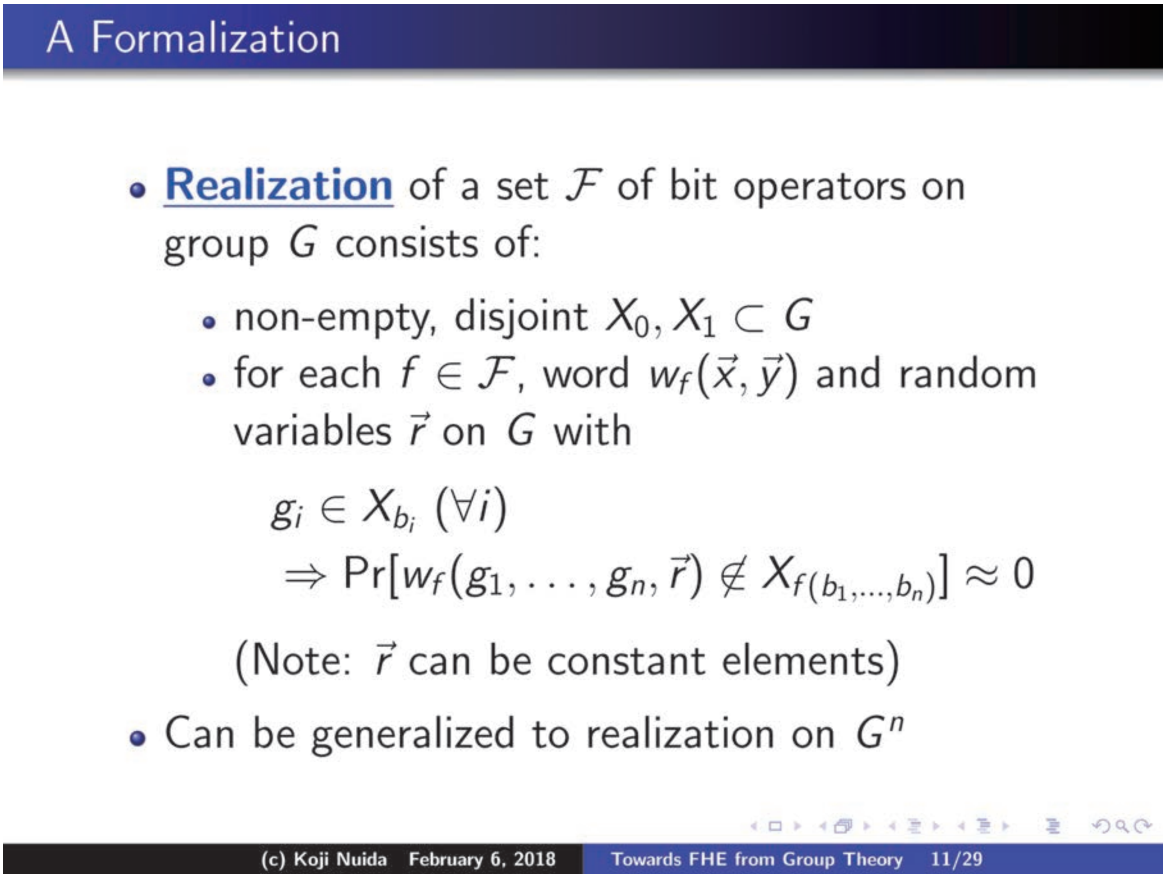
- Theorem: NAND can be "realized" on any non-commutative finite simple groups G
- Then FHE will be obtained once G can be homomorphically encrypted
  - However, no concrete way is proposed
- My research goes along this direction

Contents

- Background & problem setting
- Realization of bit operators on groups
- Detailed problem setting
- First approach (ongoing)
- Second approach (ongoing)

Preliminaries

- **Words**: sequences of variables x_i, ${x_i}^{-1}$
  - E.g., $w(x_1, x_2) = x_1 {x_2}^2 {x_1}^{-3}$
- **Substituting** group elements into a word yields a group element
  - E.g., $w(g_1, g_2) = g_1 {g_2}^2 {g_1}^{-3}$



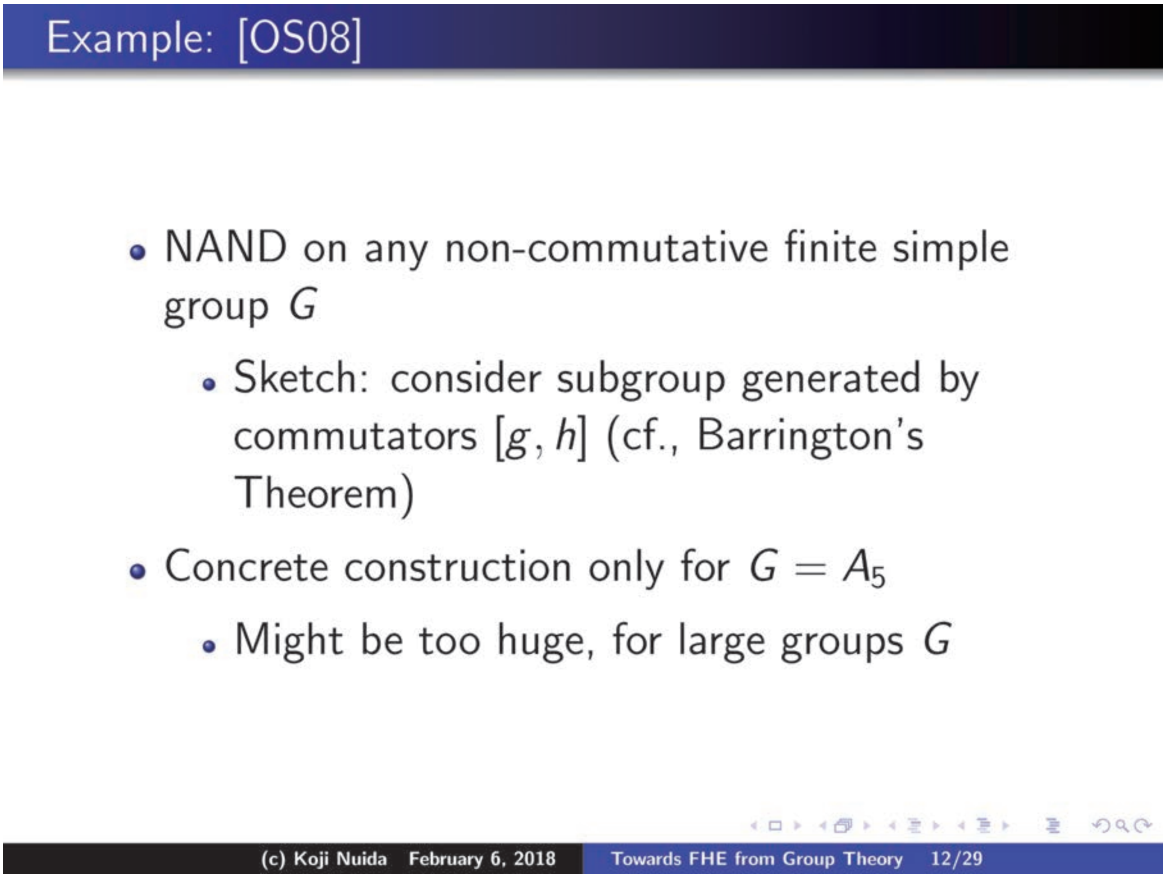
A Formalization

- **Realization** of a set $\mathcal{F}$ of bit operators on group G consists of:
  - non-empty, disjoint $X_0, X_1 \subset G$
  - for each $f \in \mathcal{F}$, word $w_f(\vec{x}, \vec{y})$ and random variables $\vec{r}$ on G with

$$g_i \in X_{b_i} \ (\forall i)$$
$$\Rightarrow \Pr[w_f(g_1, \ldots, g_n, \vec{r}) \notin X_{f(b_1,\ldots,b_n)}] \approx 0$$

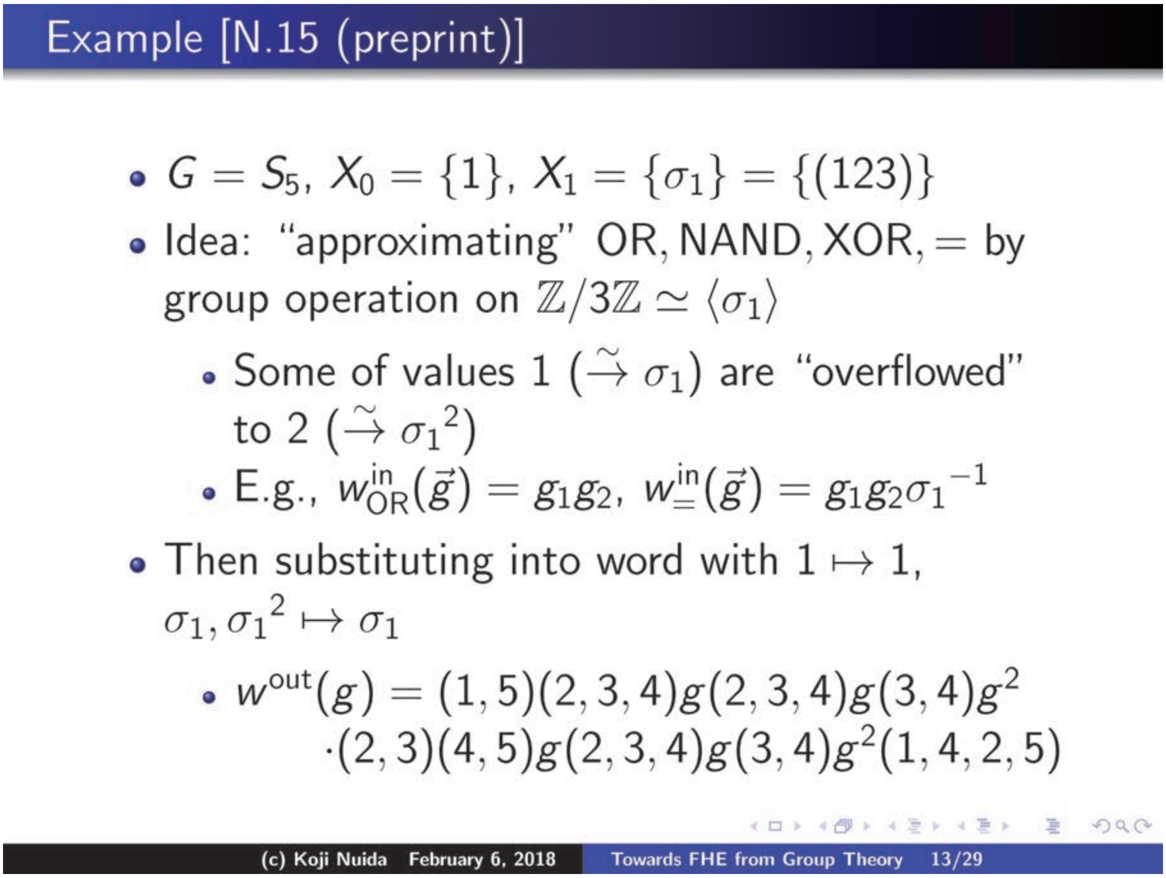
(Note: $\vec{r}$ can be constant elements)

- Can be generalized to realization on G^n



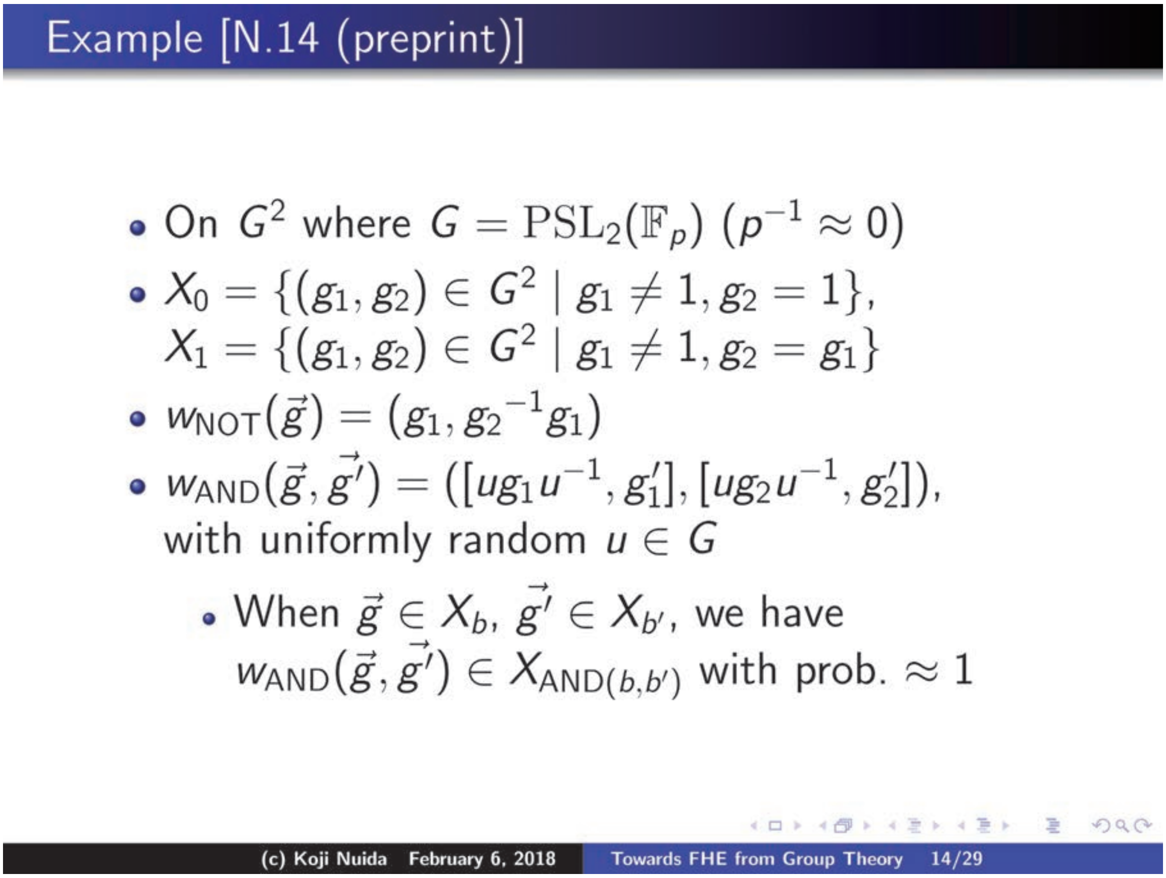
Example: [OS08]

- NAND on any non-commutative finite simple group G
  - Sketch: consider subgroup generated by commutators $[g, h]$ (cf., Barrington's Theorem)
- Concrete construction only for $G = A_5$
  - Might be too huge, for large groups G



Example [N.15 (preprint)]

- $G = S_5$, $X_0 = \{1\}$, $X_1 = \{\sigma_1\} = \{(123)\}$
- Idea: "approximating" OR, NAND, XOR, = by group operation on $\mathbb{Z}/3\mathbb{Z} \simeq \langle \sigma_1 \rangle$
  - Some of values 1 ($\overset{\sim}{\to} \sigma_1$) are "overflowed" to 2 ($\overset{\sim}{\to} {\sigma_1}^2$)
  - E.g., $w^{\mathrm{in}}_{\mathrm{OR}}(\vec{g}) = g_1 g_2$, $w^{\mathrm{in}}_{=}(\vec{g}) = g_1 g_2 {\sigma_1}^{-1}$
- Then substituting into word with $1 \mapsto 1$, $\sigma_1, {\sigma_1}^2 \mapsto \sigma_1$
  - $w^{\mathrm{out}}(g) = (1,5)(2,3,4)g(2,3,4)g(3,4)g^2$
    $\cdot(2,3)(4,5)g(2,3,4)g(3,4)g^2(1,4,2,5)$



Example [N.14 (preprint)]

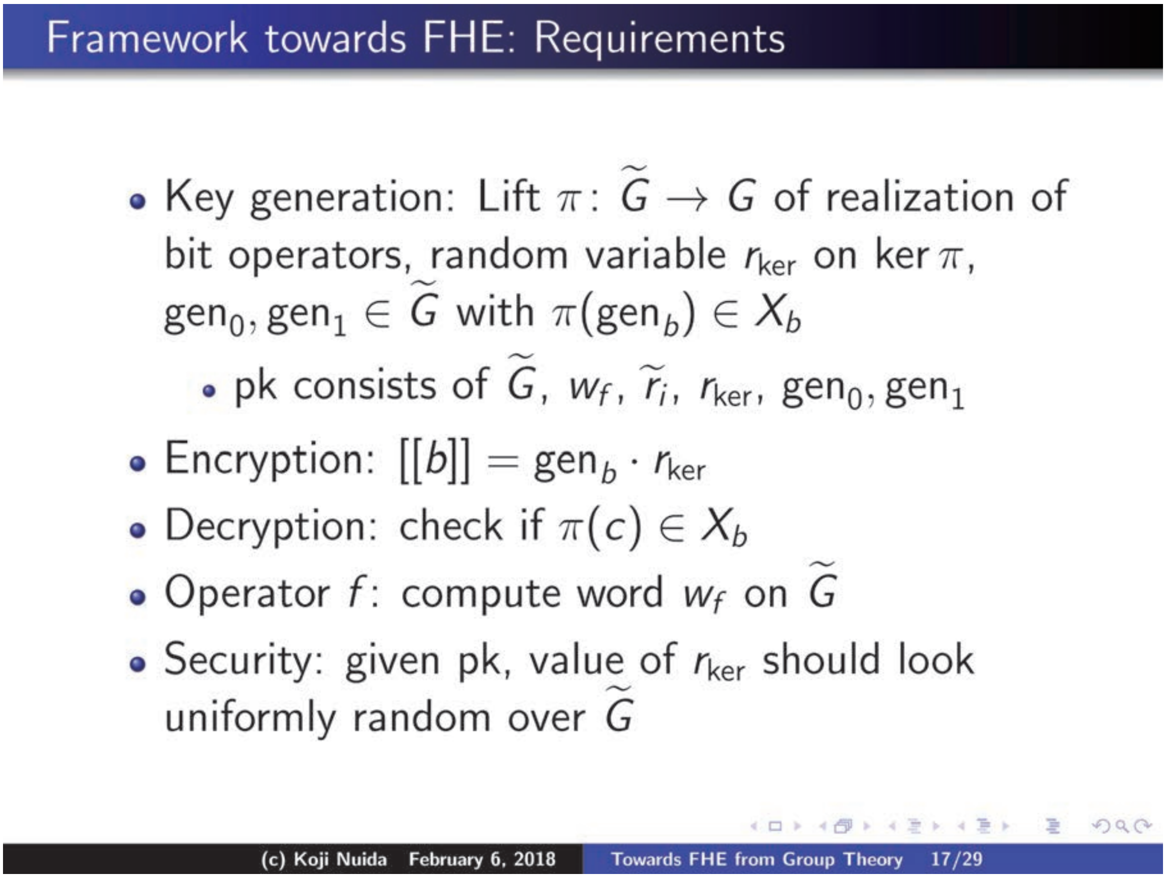
- On G^2 where $G = \mathrm{PSL}_2(\mathbb{F}_p)$ ($p^{-1} \approx 0$)
- $X_0 = \{(g_1, g_2) \in G^2 \mid g_1 \neq 1, g_2 = 1\}$,
  $X_1 = \{(g_1, g_2) \in G^2 \mid g_1 \neq 1, g_2 = g_1\}$
- $w_{\mathrm{NOT}}(\vec{g}) = (g_1, {g_2}^{-1} g_1)$
- $w_{\mathrm{AND}}(\vec{g}, \vec{g'}) = ([u g_1 u^{-1}, g'_1], [u g_2 u^{-1}, g'_2])$, with uniformly random $u \in G$
  - When $\vec{g} \in X_b$, $\vec{g'} \in X_{b'}$, we have $w_{\mathrm{AND}}(\vec{g}, \vec{g'}) \in X_{\mathrm{AND}(b,b')}$ with prob. ≈ 1

Contents

- Background & problem setting
- Realization of bit operators on groups
- Detailed problem setting
- First approach (ongoing)
- Second approach (ongoing)

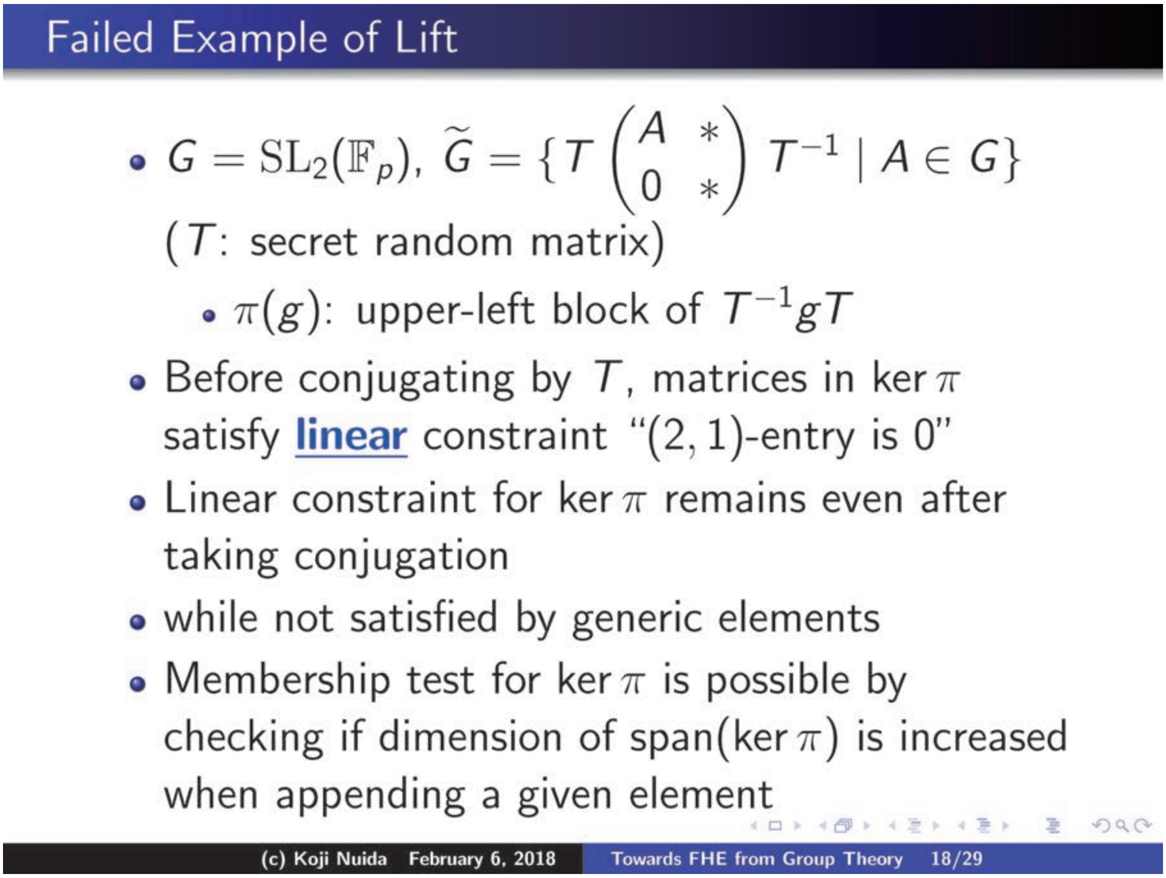
Lift of Realization of Bit Operators

- Group homomorphism $\pi\colon \widetilde{G} \to G$
- Random variables $\widetilde{r}_i$ where $\pi(\widetilde{r}_i) = r_i$ as distributions
- Then for $\pi(\widetilde{g}_i) = g_i \in X_{b_i}$, we have $\pi(w_f(\widetilde{g}_1, \dots, \widetilde{g}_n, \widetilde{r}_1, \dots, \widetilde{r}_k))$ $= w_f(g_1, \dots, g_n, r_1, \dots, r_k) \in X_{f(b_1,\dots,b_n)}$ with prob. ≈ 1



Framework towards FHE: Requirements

- Key generation: Lift $\pi\colon \widetilde{G} \to G$ of realization of bit operators, random variable r_{ker} on $\ker\pi$, $\mathrm{gen}_0, \mathrm{gen}_1 \in \widetilde{G}$ with $\pi(\mathrm{gen}_b) \in X_b$
  - pk consists of $\widetilde{G}$, w_f, $\widetilde{r}_i$, r_{ker}, $\mathrm{gen}_0, \mathrm{gen}_1$
- Encryption: $[[b]] = \mathrm{gen}_b \cdot r_{\mathrm{ker}}$
- Decryption: check if $\pi(c) \in X_b$
- Operator f: compute word w_f on $\widetilde{G}$
- Security: given pk, value of r_{ker} should look uniformly random over $\widetilde{G}$



Failed Example of Lift

- $G = \mathrm{SL}_2(\mathbb{F}_p)$, $\widetilde{G} = \{T \begin{pmatrix} A & * \\ 0 & * \end{pmatrix} T^{-1} \mid A \in G\}$ (T: secret random matrix)
  - $\pi(g)$: upper-left block of $T^{-1}gT$
- Before conjugating by T, matrices in $\ker\pi$ satisfy **linear** constraint "$(2,1)$-entry is 0"
- Linear constraint for $\ker\pi$ remains even after taking conjugation
- while not satisfied by generic elements
- Membership test for $\ker\pi$ is possible by checking if dimension of $\mathrm{span}(\ker\pi)$ is increased when appending a given element

Contents

- Background & problem setting
- Realization of bit operators on groups
- Detailed problem setting
- First approach (ongoing)
- Second approach (ongoing)

Preliminaries: Generator-Relator Presentation of Groups

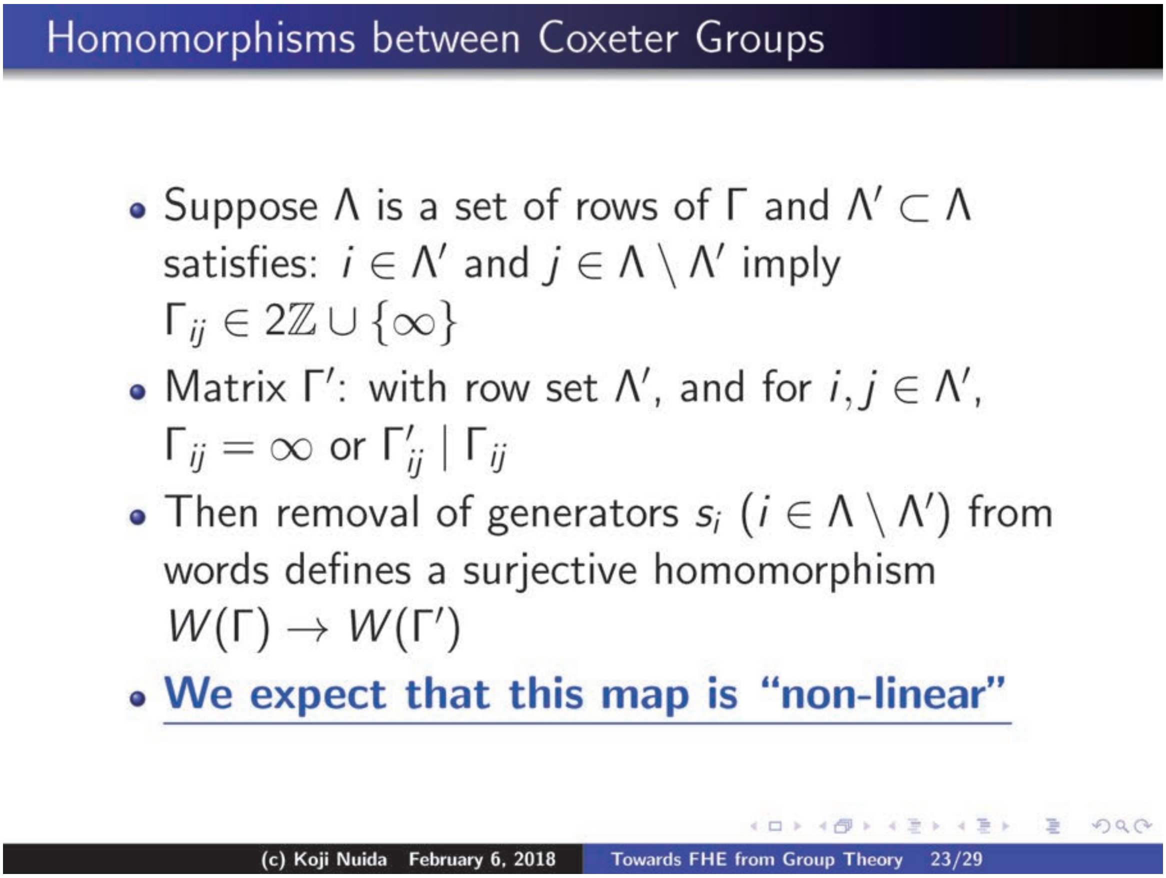
- S: set, R: set of words on S
- Group $\langle S \mid R \rangle$: quotient of words on S by "words in R are equivalent to empty word"
  - Multiplication: concatenation of words
- E.g., $S_n = \langle s_1, \dots, s_{n-1}$
  $\mid {s_i}^2, (s_i s_{i+1})^3\ (\forall i),\ (s_i s_j)^2\ (\forall i \neq j)\rangle$
  - generator s_i is transposition $(i, i+1)$

Coxeter Groups

- Coxeter matrix Γ: $\Gamma_{ii} = 1$, $\Gamma_{ij} = \Gamma_{ji} \in \{2, 3, \dots\} \cup \{\infty\}$
- Coxeter group $W(\Gamma)$: generating set S consists of rows of Γ, relations (elements of R) are $(s_i s_j)^{\Gamma_{ij}}$ $(i \leq j, \Gamma_{ij} \neq \infty)$
  - E.g., Type A_n: $\Gamma_{i,i+1} = 3$, $\Gamma_{i,j} = 2$ $(|i - j| \geq 2)$
  - i.e., S_n is a Coxeter group of type A_{n-1}

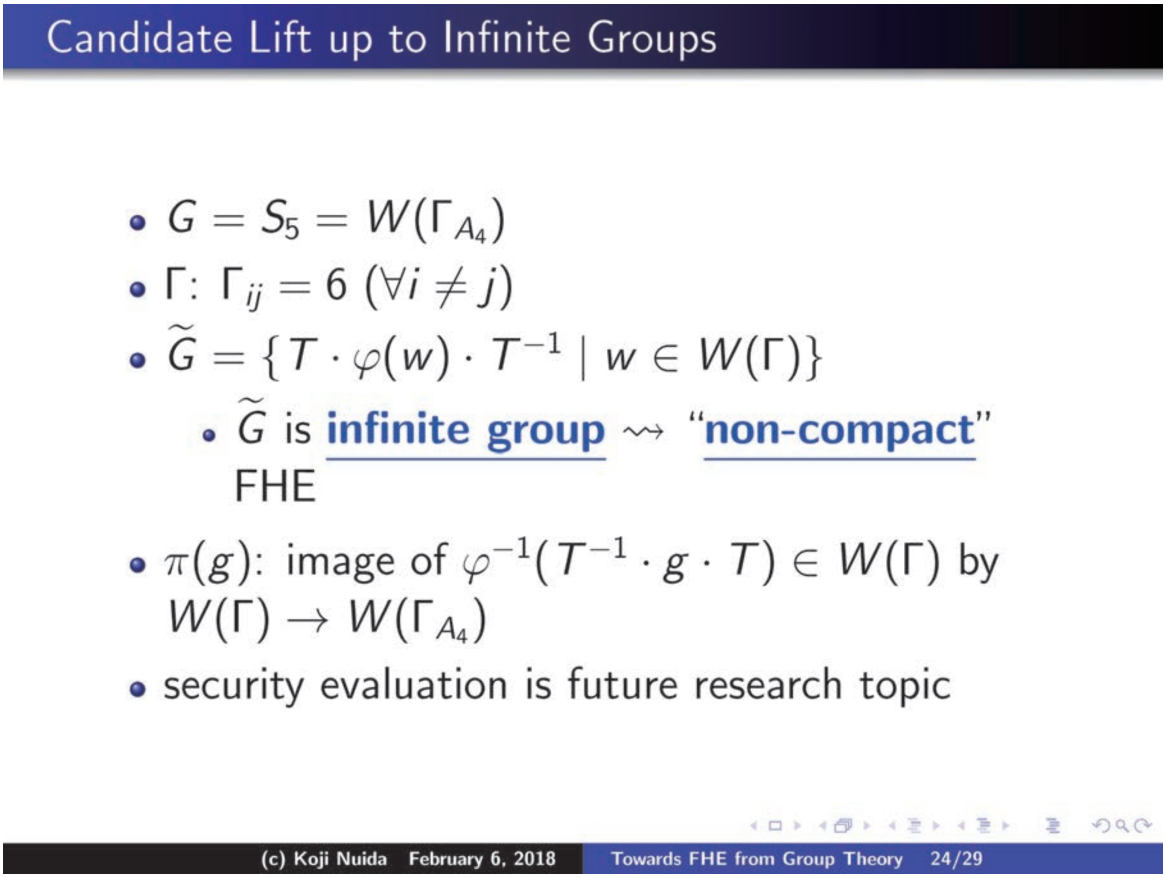
Matrix Representation for Coxeter Groups

- Generator s_j acts on unit vectors α_i by $s_j \cdot \alpha_i = \alpha_i + 2\cos(\pi/\Gamma_{ij})\alpha_j$ (where $\cos(\pi/\infty) = \cos(0) = 1$)
  - yielding matrix representation $\varphi(w)$ of $w \in W(\Gamma)$
- φ: $W(\Gamma) \to \mathrm{GL}_n(\mathbb{R})$ is **injective**
- recursive computation of φ^{-1}: if i-th column of $\varphi(w)$ has negative entry, then ws_i is shorter than w



Homomorphisms between Coxeter Groups

- Suppose Λ is a set of rows of Γ and $\Lambda' \subset \Lambda$ satisfies: $i \in \Lambda'$ and $j \in \Lambda \setminus \Lambda'$ imply $\Gamma_{ij} \in 2\mathbb{Z} \cup \{\infty\}$
- Matrix Γ': with row set Λ', and for $i, j \in \Lambda'$, $\Gamma_{ij} = \infty$ or $\Gamma'_{ij} \mid \Gamma_{ij}$
- Then removal of generators s_i $(i \in \Lambda \setminus \Lambda')$ from words defines a surjective homomorphism $W(\Gamma) \to W(\Gamma')$
- **We expect that this map is "non-linear"**



Candidate Lift up to Infinite Groups

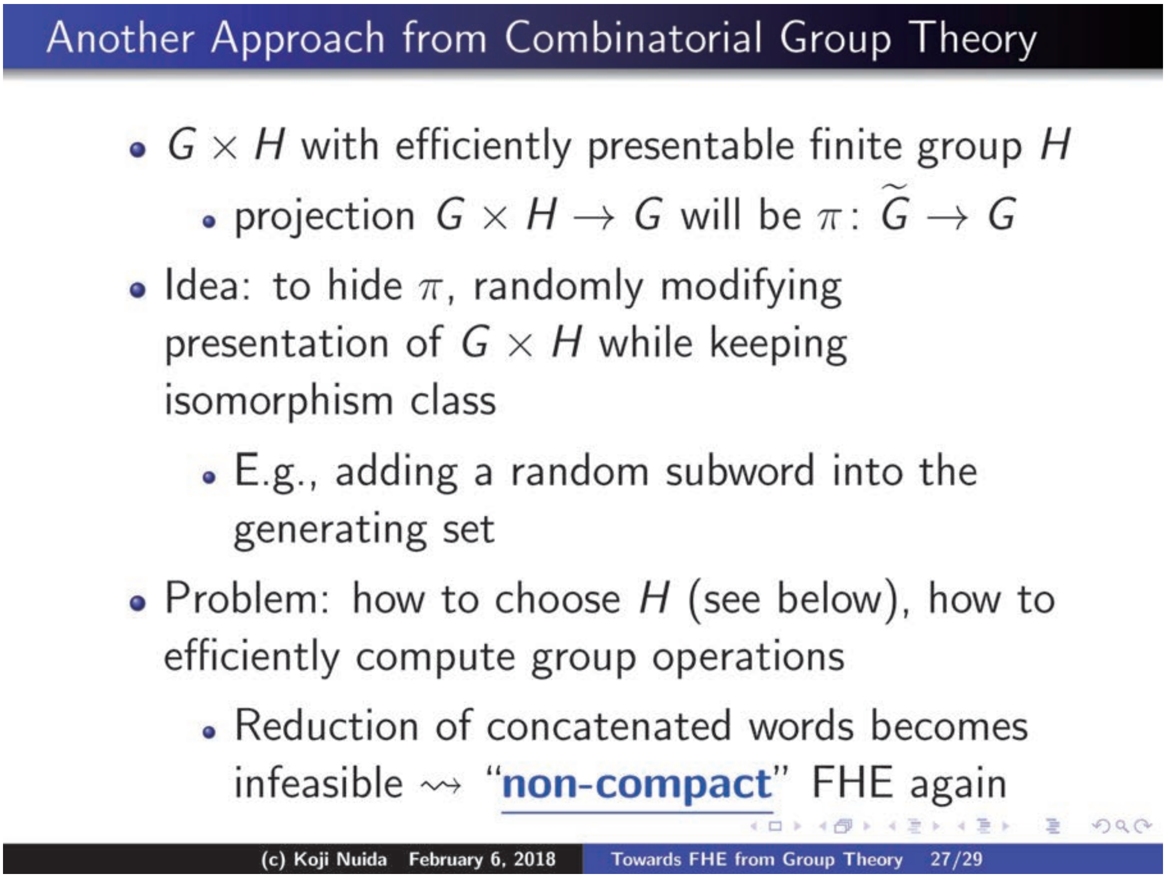
- $G = S_5 = W(\Gamma_{A_4})$
- Γ: $\Gamma_{ij} = 6$ $(\forall i \neq j)$
- $\widetilde{G} = \{T \cdot \varphi(w) \cdot T^{-1} \mid w \in W(\Gamma)\}$
  - $\widetilde{G}$ is **infinite group** $\rightsquigarrow$ "**non-compact**" FHE
- $\pi(g)$: image of $\varphi^{-1}(T^{-1} \cdot g \cdot T) \in W(\Gamma)$ by $W(\Gamma) \to W(\Gamma_{A_4})$
- security evaluation is future research topic

Question: Why Not Using Finite Groups?

- Only "irreducible" Γ yielding $W(\Gamma) \to W(\Gamma_{A_n})$ $(n \geq 4)$ are of type B_{n+1}
  - $\Gamma_{B_{n+1}} = \begin{pmatrix} & & & & 2 \\ & & & & \vdots \\ & & \Gamma_{A_n} & & 2 \\ & & & & 4 \\ 2 & \cdots & 2 & 4 & 1 \end{pmatrix}$
- Then matrices in ker are lower-triangular, hence **linear** constraint "upper entries are 0" remains
- Groups other than Coxeter groups needed

Contents

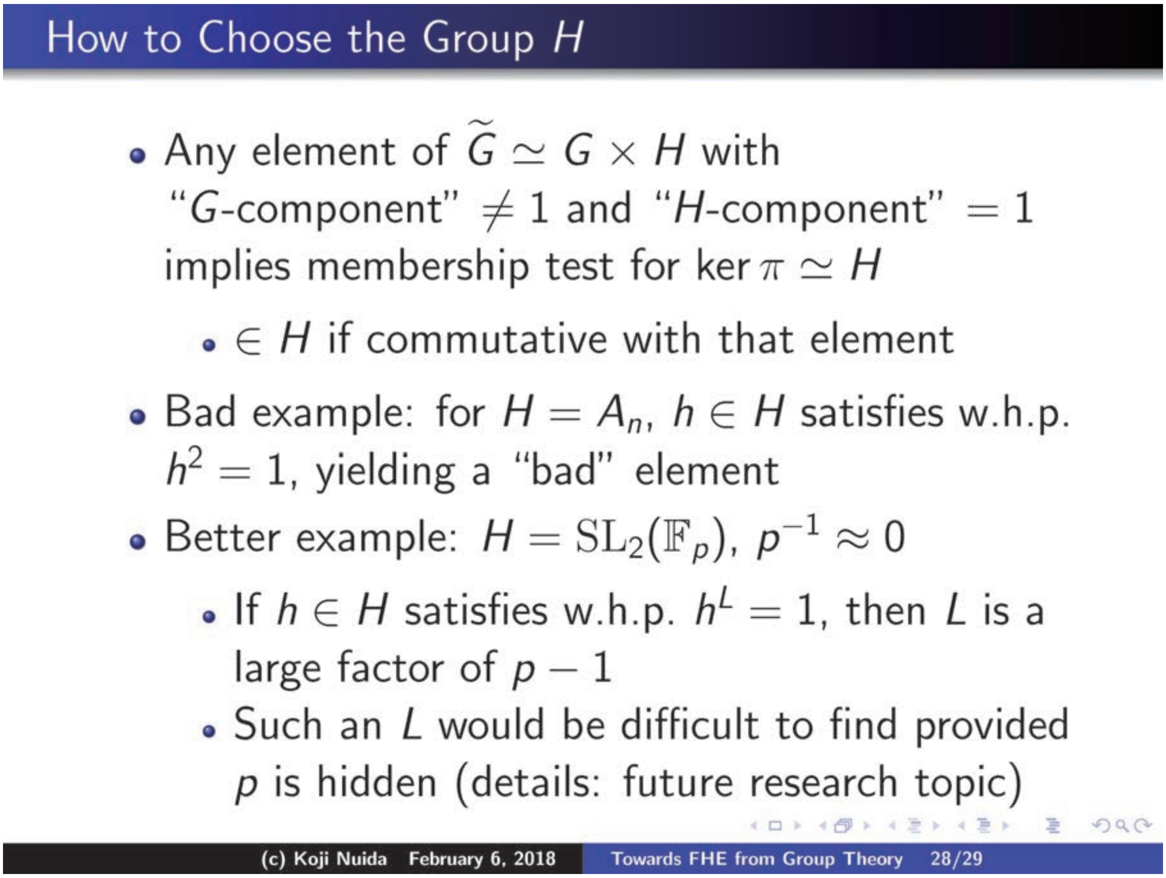
- Background & problem setting
- Realization of bit operators on groups
- Detailed problem setting
- First approach (ongoing)
- Second approach (ongoing)



Another Approach from Combinatorial Group Theory

- $G \times H$ with efficiently presentable finite group H
  - projection $G \times H \to G$ will be $\pi \colon \widetilde{G} \to G$
- Idea: to hide π, randomly modifying presentation of $G \times H$ while keeping isomorphism class
  - E.g., adding a random subword into the generating set
- Problem: how to choose H (see below), how to efficiently compute group operations
  - Reduction of concatenated words becomes infeasible $\rightsquigarrow$ "**non-compact**" FHE again

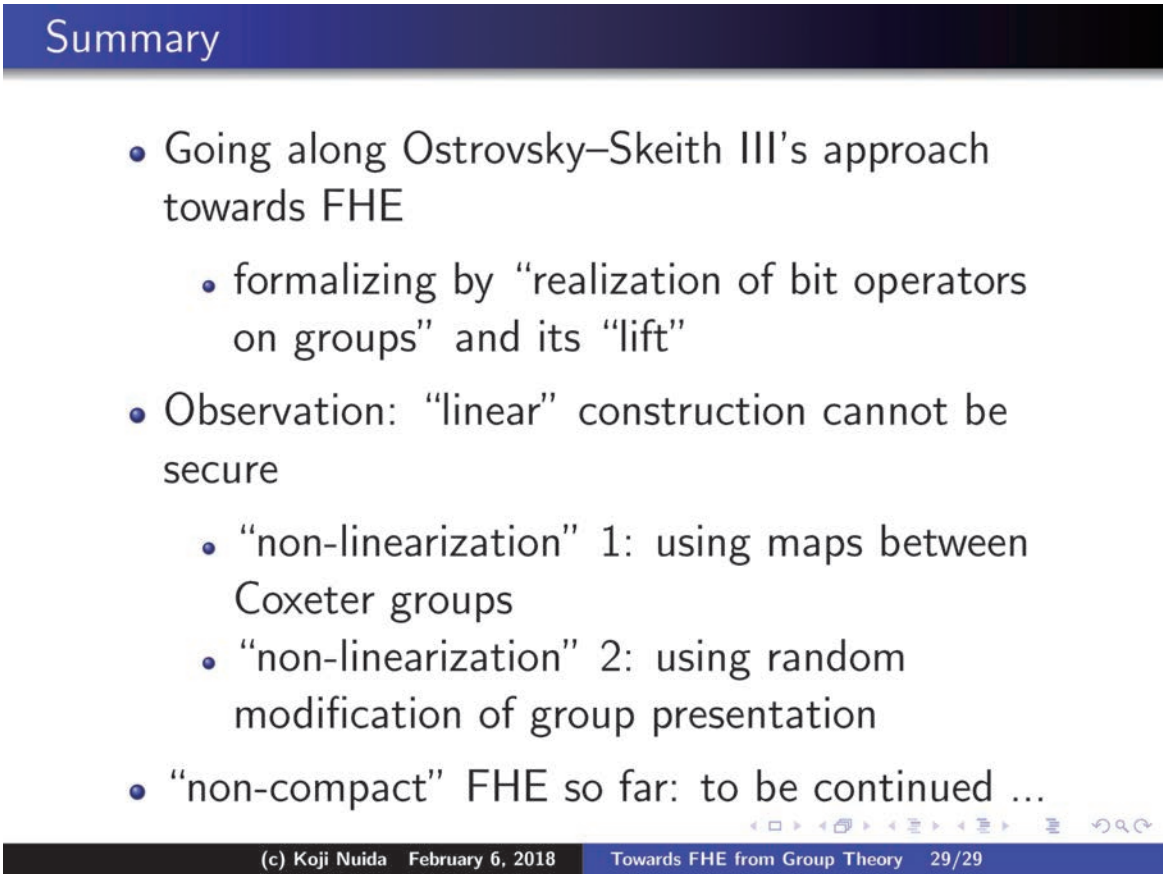
(c) Koji Nuida February 6, 2018 Towards FHE from Group Theory 27/29



How to Choose the Group H

- Any element of $\widetilde{G} \simeq G \times H$ with "G-component" $\neq 1$ and "H-component" $= 1$ implies membership test for $\ker \pi \simeq H$
  - $\in H$ if commutative with that element
- Bad example: for $H = A_n$, $h \in H$ satisfies w.h.p. $h^2 = 1$, yielding a "bad" element
- Better example: $H = \mathrm{SL}_2(\mathbb{F}_p)$, $p^{-1} \approx 0$
  - If $h \in H$ satisfies w.h.p. $h^L = 1$, then L is a large factor of $p - 1$
  - Such an L would be difficult to find provided p is hidden (details: future research topic)

(c) Koji Nuida February 6, 2018 Towards FHE from Group Theory 28/29



Summary

- Going along Ostrovsky–Skeith III's approach towards FHE
  - formalizing by "realization of bit operators on groups" and its "lift"
- Observation: "linear" construction cannot be secure
  - "non-linearization" 1: using maps between Coxeter groups
  - "non-linearization" 2: using random modification of group presentation
- "non-compact" FHE so far: to be continued ...

A survey on Multivariate Public Key Cryptosystem

Yasufumi Hashimoto (Univ. Ryukyu)

Multivariate Public Key Cryptosystem, MPKC

A public key cryptosytem whose public key is a set of multivariate quadratic polynomials over a finite field.

$$\begin{aligned} f_1(x_1, \cdots, x_n) &= \sum_{1 \le i \le j \le n} a_{ij}^{(1)} x_i x_j + \sum_{1 \le i \le n} b_i^{(1)} x_i + c^{(1)}, \\ &\vdots \\ f_m(x_1, \cdots, x_n) &= \sum_{1 \le i \le j \le n} a_{ij}^{(m)} x_i x_j + \sum_{1 \le i \le n} b_i^{(m)} x_i + c^{(m)}. \end{aligned}$$

For a plain-text $(x_1, \ldots, x_n)$, the cipher-text $(y_1, \ldots, y_m)$ is given by

$$y_1 = f_1(x_1, \cdots, x_n), \quad \ldots, \quad y_m = f_m(x_1, \cdots, x_n).$$

It is, in general, too difficult to find a solution of

$$f_1(x) = y_1, \quad \cdots, \quad f_m(x) = y_m$$

for randomly chosen quadratic forms $f_1, \dots, f_m$ (NP-hard).

It has been expected that MPKC will be one of *Post-Quantum Cryptographies*.

The encryption is efficient.

RSA: $y \equiv x^e \bmod n$,
(x, y, e, n are hundreds or thousands bits).
MPKC: $y = F(x)$,
(q is $2 \sim 257$?, n, m are $40 \sim 200$?).

Matsumoto-Imai's Cryptosystem (Eurocrypt'88)

$n \geq 1$,
k: a finite field of even char., $q := \#k$,
K: an n-extension of k,
$\phi : k^n \to K$: a one-to-one map,
$\mathcal{G} : K \to K$:

$$\mathcal{G}(X) = X^{1+q^l}, \qquad (l \geq 1, \gcd(q^n - 1, q^l + 1) = 1).$$

Secret key: $S, T : k^n \to k^n$: invertible affine (or linear) maps,
Public key: $F := T \circ \phi^{-1} \circ \mathcal{G} \circ \phi \circ S$.

$$F : k^n \xrightarrow{S} k^n \xrightarrow{\phi} K \xrightarrow{\mathcal{G}} K \xrightarrow{\phi^{-1}} k^n \xrightarrow{T} k^n$$

F is a quadratic map.

$$
\begin{aligned}
X =& x_1\theta_1 + \cdots + x_n\theta_n, && ((\theta_1,\ldots,\theta_n) \text{ is a } k\text{-basis of } K)\\
X^{q^l} =& x_1\theta_1^{q^l} + \cdots + x_n\theta_n^{q^l} && ((a+b)^q = a^q + b^q,\ x_i^q = x_i)\\
=& (x_1,\ldots,x_n\text{-linear})\cdot\theta_1 + \cdots + (x_1,\ldots,x_n\text{-linear})\cdot\theta_n.
\end{aligned}
$$

Encryption: For a plain-text $x \in k^n$, the cipher $y \in k^n$ is

$$y = F(x).$$

Decryption:

$$x = S^{-1}(\phi^{-1}(\phi(T^{-1}(y))^{N})),$$

where N is an integer with $(1+q^l)N \equiv 1 \bmod q^n - 1$, namely $(X^{q^l+1})^N = X$.

Moon Letter Cryptosystem (Tsujii-Kurosawa-Itoh-Fujioka-Matsumoto, 1986)

k: a finite field,
$q := \#k$,
$n \geq 1$,
$x = (x_1,\ldots,x_n)^t$,
$G(x) = (g_1(x),\ldots,g_n(x))^t$: a quadratic map defined by

$$
\begin{aligned}
g_1(x) =& (x_1\text{-linear form}),\\
g_2(x) =& x_2\cdot(x_1\text{-linear form}) + (x_1\text{-quadratic form}),\\
g_3(x) =& x_3\cdot(x_1, x_2\text{-linear form}) + (x_1, x_2\text{-quadratic form}),\\
&\vdots\\
g_n(x) =& x_n\cdot(x_1,\ldots,x_{n-1}\text{-linear form}) + (x_1,\ldots,x_{n-1}\text{-quadratic form}).
\end{aligned}
$$

Secret key: $S, T : k^n \to k^n$: invertible affine (or linear) maps,

Public key: $F := T \circ G \circ S$.

Encryption: For a plain-text $x \in k^n$, the cipher-text is $y = F(x)$.

Decryption: For $z = (z_1, \dots, z_n)^t := T^{-1}(y)$, first find x_1 with

$$g_1(x) = z_1, \qquad (x_1\text{-linear equation})$$

and substitute it into other polynomials. Next, find x_2 with

$$g_2(x) = z_2, \qquad (x_2\text{-linear equation})$$

and substitute it into other polynomials. Contitue it and find $x_3, \dots, x_n$ recursively.
Finally, compute $S^{-1}(x_1, \dots, x_n)^t$, which is the plain-text.

Remark: Later in Crypto'93, Shamir proposed a similar scheme almost same to it.

General construction

Most MPKCs are constructed as follows.

$n \geq 1$: the number of variables.
$m \geq 1$: the number of quadratic forms.
k: a finite field, $q := \#k$.

Secret key.

$S : k^n \to k^n$, an invertible affine (or linear) map.
$G : k^n \to k^m$, a quadratic map "inverted feasibly".
$T : k^m \to k^m$, an invertible affine (or linear) map.

Public key.

$F := T \circ G \circ S : k^n \to k^m$.

Encryption.

For a plain-text $x \in k^n$, the cipher-text is $y = F(x) \in k^m$.

Decryption.

The plain-text x is given by $x = S^{-1}(G^{-1}(T^{-1}(y)))$.

G: a quadratic map inverted feasibly.

$$\begin{pmatrix} f_1(x) \\ \vdots \\ f_m(x) \end{pmatrix} = T \begin{pmatrix} g_1(Sx) \\ \vdots \\ g_m(Sx) \end{pmatrix}$$

F: a quadratic map, not inverted feasibly?

Q. Is it really secure **?**

A. Not necessarily.

An attack on MI (Patarin, Crypto'95)

Since the encryption is by $Y = X^{1+q^l}$, it holds

$$YX^{q^{2l}} = Y^{q^l}X \left(= X^{1+q^l+q^{2l}}\right).$$

Y, Y^{q^l}: linear forms of the cipher-text $y = (y_1, \dots, y_n)$,
$X, X^{q^{2l}}$: linear forms of the plain-text $x = (x_1, \dots, x_n)$.

There exist polynomial equations in the forms

$$\sum_{1 \le i,j \le n} \alpha_{ij} x_i y_j + \sum_{1 \le i \le n} \beta_i x_i + \sum_{1 \le j \le n} \gamma_j y_j + \delta = 0,$$

which hold for any plaintext-ciphertext pairs (x, y).

Generate (x, y) sufficiently many and find such polynomials. □

Attack on ML (Hasegawa-Kaneko, 1987, Coppersmith-Sten-Vaudenay, Crypto'93)

The coefficient matrices $G_1, \dots, G_n$ of $g_1(x), \dots, g_n(x)$ (i.e. $g_i(x) = x^t G_i x + (\text{linear})$) are written by

$$G_n = \begin{pmatrix} \ast_{n-1} & \ast \\ \ast & 0 \end{pmatrix}, \qquad G_{n-1} = \begin{pmatrix} \ast_{n-1} & 0 \\ 0 & 0 \end{pmatrix}, \qquad \cdots .$$

Since the public quadratic forms are linear sums of linear transforms of the aboves, there exists $\alpha \in k$ such that

$$\mathrm{rank}(F_1 - \alpha F_2) \leq n - 1 \Leftrightarrow \det(F_1 - \alpha F_2) = 0.$$

This α is a part of T. Once such an α is found, S is recovered partially.
After that, we can recover further information of S, T recursively. □

There are many G's generating insecure MPKCs.

Q. Which kind of G generates a secure MPKCs?

A. There are no schemes with provable security.
We know several properties of G to be broken.

We give several major attacks on MPKCs.

Major Attacks on MPKCs

1. Direct attacks.

It is to solve the system of multivariate quadratic equations

$$\begin{aligned} f_1(x_1,\ldots,x_n) &= y_1, \\ f_2(x_1,\ldots,x_n) &= y_2, \\ &\vdots \\ f_m(x_1,\ldots,x_n) &= y_m \end{aligned}$$

directly to recover the plain-text $x=(x_1,\ldots,x_n)$.

Exhaustive search: $O(q^{\min(m,n)}\cdot(\text{polyn.}))$.

Grover's (quantum) algorithm: $O(q^{\frac{1}{2}\min(m,n)}\cdot(\text{polyn.}))$.

Gröbner basis algorithm.

Buchberger's algorithm: $O\left(2^{2^n}\right)$.

$\xrightarrow{\text{Improve}}$ F_4-, F_5-algorithms (Faugere, 2001$\sim$):

If $n=m$, the complexity seems $O(A^m)$ where $A\sim 10$.

If $m\gg n$ (over-defined), it is more efficient.

Especially if $m\geq\frac{1}{2}n(n+1)$, it solves in polynomial time (if a solution exists).

$\because \quad A\left(x_1^2,x_1x_2,\cdots,x_n^2,x_1,\cdots,x_n\right)^t=b. \quad \square$

If $n \gg m$ (under-defined), there are efficient algorithms.

Especially if $n \geq \frac{1}{2}m(m+1)$, we can find a solution in polynomial time (H, Miura-H-Takagi, Cheng-H-M-T, 2009∼2014).

$$\begin{cases} f_1(x) = y_1 \\ f_2(x) = y_2 \\ \quad \vdots \\ f_m(x) = y_m \end{cases} \xrightarrow{\text{linear transf.}} \begin{cases} (x_1\text{-quadratic}) = 0 \\ (x_1, x_2\text{-quadratic}) = 0 \\ \quad \vdots \\ (x_1, \ldots, x_m\text{-quadratic}) = 0 \end{cases}$$

Even if $n < \frac{1}{2}m(m+1)$, there are efficient algorithms by combining the Gröbner basis algorithm.

$$\begin{array}{c|l} n \geq \frac{1}{2}m(m+1) & \text{polynomial time} \\ \vdots & \quad \uparrow \text{ more efficient} \\ n = m & \text{almost } O(10^n) \\ \vdots & \quad \downarrow \text{ more efficient} \\ \frac{1}{2}n(n+1) \leq m & \text{polynomial time} \end{array}$$

2. Rank attacks.

To recover T partially by checking the ranks of coefficient matrices of the quadratic forms in F and G.

$G_1, \dots, G_m$: $n \times n$ matrices with $g_l(x) = x^t G_l x + (\text{linear})$.
$F_1, \dots, F_m$: $n \times n$ matrices with $f_l(x) = x^t F_l x + (\text{linear})$.

$$F_j = \sum_{1 \le i \le m} t_{ij}(S^t G_i S) = S^t \left(\sum_{1 \le i \le m} t_{ij} G_i \right) S, \qquad (T = (t_{ij})).$$

Example. ML.

G_n: rank n, G_{n-1}: rank $n-1$, ...
$\Rightarrow \exists \alpha \in k$ s.t. $\mathrm{rank}(F_1 - \alpha F_2) \le n-1$.

Min-rank attack: $\exists \alpha_1, \dots, \alpha_m \in k$ and $R \ge 1$ s.t.

$$\mathrm{rank}(\alpha_1 F_1 + \cdots + \alpha_m F_m) \le R.$$

If R is smaller, the min-rank attack is more efficient.

High-rank attack: $\exists R, L \ge 1$ and $\beta_1, \dots, \beta_L \in k$ s.t.

$$\mathrm{rank}(F_m - \beta_1 F_1 - \cdots - \beta_L F_L) \le R.$$

If L is smaller, the high-rank attack is more efficient.

When q *is small enough*, we can find $\{\alpha_i\}$ or $\{\beta_i\}$ *exhaustively*. The complexity is

MR: $(q^R \cdot (\text{polyn.}))$, quantum: $(q^{R/2} \cdot (\text{polyn.}))$,
HR: $(q^L \cdot (\text{polyn.}))$, quantum: $(q^{L/2} \cdot (\text{polyn.}))$.

When q *is large*, generate a system of polynomial equations derived from the condition for the rank and solve it. It is

MR: m variables, degree $R+1$,
HR: L variables, degree $R+1$.

3. Conjugation attack.

When the coefficient matrices $G_1, \ldots, G_m$ are in special forms, recover S by using its conjugation properties.

$$F_j = S^t H_j S, \quad (H_j \text{ is a linear sum of } G_1, \ldots, G_m),$$
$$\Rightarrow \qquad F_1^{-1} F_2 = S^{-1}(H_1^{-1} H_2) S.$$

If $H_1^{-1} H_2$ is in special form, one can recover S easily.

Ex. 1. An attack on Oil-Vinegar signature scheme (Kipnis-Shamir, 1997)

$$G_1, \dots, G_m = \begin{pmatrix} 0_m & * \\ * & *_m \end{pmatrix},$$

The inversions of these matrices are $\begin{pmatrix} *_m & * \\ * & 0_m \end{pmatrix}$.

$$\Rightarrow F_1^{-1}F_2 = S^{-1} \begin{pmatrix} *_m & * \\ 0 & *_m \end{pmatrix} S$$

Find S_1 s.t. $S_1^{-1}(F_1^{-1}F_2)S_1 = \begin{pmatrix} *_m & * \\ 0 & *_m \end{pmatrix}$.

Then S_1 satisfies $SS_1 = \begin{pmatrix} *_m & * \\ 0 & *_m \end{pmatrix}$.

Ex. 2. An attack on Multi-HFE (H, 2015)

$$\tilde{G}_1, \dots, \tilde{G}_m = \begin{pmatrix} *_N & & \\ & \ddots & \\ & & *_N \end{pmatrix},$$

The inversions of these matrices are same.

$$\Rightarrow F_1^{-1}F_2 = \tilde{S}^{-1} \begin{pmatrix} *_N & & \\ & \ddots & \\ & & *_N \end{pmatrix} \tilde{S}$$

After diagonalizing $F_1^{-1}F_2$,

one can recover S_1 s.t. $\tilde{S}S_1 = \begin{pmatrix} *_N & & \\ & \ddots & \\ & & *_N \end{pmatrix} \cdot$ (perm.).

4. Linearization attack.

An attack to recover polynomial equations

$$H(x_1, \dots, x_n, y_1, \dots, y_m) = 0,$$

if arbitrary pairs of plaintext $x = (x_1, \dots, x_n)^t$ and ciphertext $y = (y_1, \dots, y_m)^t$ satisfy them.

Prepare sufficiently many p-c pairs (x, y), substitute them in $H(x, y)$ and determine the coefficients of $H(x, y)$.

MI: H is linear in x and in y.

If the degree of H is smaller, this attack is more efficient.

CPA (Chosen Plaintext Attack).

(x, y): $x \longmapsto y$.

Available on MI.

CCA (Chosen Ciphertext Attack).

(x, y): $y \longmapsto x$.

If a special polynomial are used in the decryption process, it can be recovered by CCA.

5. Differential attack.

An attack by using the difference

$$DF(x, c) := F(x + c) - F(x) - F(c) + F(0).$$

If there exist a polynomial $H(a)$ s.t.

$$DF(ax, c) + DF(x, ac) = H(a) \cdot DF(x, c),$$

one can recover useful information to decrypt.

Sflash, selected in NESSIE (2003), was broken by Dubois-Fouque-Shamir-Stern (2007).

6. Physical attacks.

Side channel attack.

Available on Sflash (Okeya-Takagi-Vuillaume, 2005)

Fault attack.

Available on most MPKCs (H-T, 2011).

There is a simple contermeasure.

1. Direct attack: Solving a system of quadratic equations directly to recover a plain-text.

2. Rank attack: Using the property of the ranks of coefficient matrices to recover a secret key.

3. Conjugation attack: Using the conjugation property of coefficient matrices to recover a secret key.

4. Linearization attack: Recovering (non-trivial) polynomial equations by plaintext-ciphertext pairs.

5. Differential attack: Using the properties of differentials of the quadratic forms.

6. Physical attacks: Available on most MPKCs under naive implementations. There is a simple countermeasure.

etc.

Proposed MPKCs

Stepwise type schemes.

The quadratic equations are solved step-by-step.

ML.

$$\begin{aligned} g_1(x) =& (x_1\text{-linear}),\\ g_2(x) =& x_2 \cdot (x_1\text{-linear}) + (x_1\text{-quadratic}),\\ g_3(x) =& x_3 \cdot (x_1, x_2\text{-linear}) + (x_1, x_2\text{-quadratic}),\\ &\vdots\\ g_n(x) =& x_n \cdot (x_1, \dots, x_{n-1}\text{-linear}) + (x_1, \dots, x_{n-1}\text{-quadratic}). \end{aligned}$$

Broken by the high-rank attack.

Oil-Vinegar signature scheme (Patarin, 1997).

$n = 2m$ (under-defined).

$$g_1(x), \ldots, g_m(x) = \sum_{1 \le i \le m} x_i \cdot (x_{m+1}, \ldots, x_{2m}\text{-linear}) + (x_{m+1}, \ldots, x_{2m}\text{-quadratic}).$$

Signature generation.

1. Choose $u_1, \ldots, u_m \in k$ randomly.
2. Solve a system of linear equations

$$\begin{aligned} g_1(x_1, \ldots, x_m, u_1, \ldots, u_m) &= y_1, \\ &\vdots \\ g_m(x_1, \ldots, x_m, u_1, \ldots, u_m) &= y_m. \end{aligned}$$

Transform $(x_1, \ldots, x_m, u_1, \ldots, u_m)^t$ by S^{-1}, which is a signature.

Since the coefficient matrices are $\begin{pmatrix} 0_m & * \\ * & *_m \end{pmatrix}$, it is broken by the conjugation attack (Kipnis-Shamir, 1997).

Unbalanced Oil-Vinegar signature scheme.

(UOV, Kipnis-Patarin-Goubin, 1999)

$n = 2m+v$, $(v \ge 1)$.

$g_1(x), \ldots, g_m(x) = x^t \begin{pmatrix} 0_m & * \\ * & *_{m+v} \end{pmatrix} x + (\text{linear form})$.

Not broken by the conjugation attack directly.
But an arranged one recovers an equivalent key in time
$O(q^v \cdot (\text{polyn.}))$

Good: Signature generation is simple, and the security seems enough under suitable parameter selection.

Not good: Key size is relatively large.

Rainbow. (Multi-layer UOV, Hybrid of ML and UOV, Ding-Schmidt, 2005)

Double-layer version.

$o_1, o_2, v \geq 1.$
$n := o_1 + o_2 + v,\ m := o_1 + o_2.$

$$\begin{aligned} g_1(x), \ldots, g_{o_1}(x) &= \sum_{1 \leq i \leq o_1} x_i(x_{o_1+1}, \ldots, x_n\text{-linear}) \\ &\quad + (x_{o_1}, \ldots, x_n\text{-quadratic}), \\ g_{o_1+1}(x), \ldots, g_m(x) &= \sum_{o_1+1 \leq i \leq m} x_i(x_{m+1}, \ldots, x_n\text{-linear}) \\ &\quad + (x_{m+1}, \ldots, x_n\text{-quadratic}). \end{aligned}$$

Signature generation: Substite random values into $x_{m+1}, \ldots, x_n$ and solve systems of linear equations step-by-step.

Coefficient matrices are as follows.

$$G_1, \ldots, G_{o_1} = \begin{pmatrix} 0_{o_1} & * & * \\ * & *_{o_2} & * \\ * & * & *_v \end{pmatrix},$$
$$G_{o_1+1}, \ldots, G_m = \begin{pmatrix} 0_{o_1} & 0 & 0 \\ 0 & 0_{o_2} & * \\ 0 & * & *_v \end{pmatrix}.$$

Security:
High-rank attack: $O(q^{o_1} \cdot (\text{polyn.}))$,
quantum: $O(q^{o_1/2} \cdot (\text{polyn.}))$.
Min-rank attack: $O(q^{o_2+v} \cdot (\text{polyn.}))$,
quantum: $O(q^{(o_2+v)/2} \cdot (\text{polyn.}))$.
Arranged conjugation attack: $O(q^{o_2+v-o_1} \cdot (\text{polyn.}))$,
quantum: $O(q^{(o_2+v-o_1)/2} \cdot (\text{polyn.}))$.

If o_1, o_2, v are similar, $n \sim 1.5m$
the security is about $O(q^o \cdot (\text{polyn.}))$
(quantum: $O(q^{o/2} \cdot (\text{polyn.}))$).

Good: The signature generation is simple and the security is enough under suitable parameter selection.
Key size is smaller than UOV.

There are many ideas to reduce the key size more.
e.g. Chen-Yang (2003), Petzold-Bulygin-Buchmann (2010∼), Yasuda-Takagi-Sakurai (2014), ...

	Linear Comp.	Security	Size
ML	Small, Many times	Bad	Small
Rainbow	Middle, Twice	(maybe) Good	Middle
UOV	Large, Once	(maybe) Good	Large

Security and the speed of signature generation are good.

The key size is relatively large.

Extension field type

Generating a quadratic map by a polynomial map over an extension field.

k: a finite field.
K: an extension field of k.
$\mathcal{G}$: a polynomial map over K.

$$G : k^n \xrightarrow{\text{1-1}} K^N \xrightarrow{\mathcal{G}} K^M \xrightarrow{\text{1-1}} k^m$$

MI (1980's)

$N = M = 1$.
$\mathcal{G}(X) = X^{1+q^l}$.

Broken by the linearization attack (Patarin, 1995).

HFE (Hidden Field Equation, Patarin, 1996)

$N = M = 1$ $(n = m)$,
$r \ll n - 1$.

$$\mathcal{G}(X) = \sum_{0 \le i \le j \le r} \alpha_{ij} X^{q^i + q^j} + \sum_{0 \le i \le r} \beta_i X^{q^i} + \gamma.$$

Decryption: Solve a univariate equation $\mathcal{G}(X) - Y = 0$.

The complexity by Berlekamp's algorithm is about $q^{4r \sim 6r}$.

$\Rightarrow$ relatively heavy!

Security: The min-rank attack is available if r is small.

$x = (x_1, \ldots, x_n)^t \mapsto (X, X^q, \ldots, X^{q^{n-1}})^t =: \tilde{X}$

$\mathcal{G}(X) = \tilde{X}^t \begin{pmatrix} *_{r+1} & \\ & \end{pmatrix} \tilde{X} + (\text{linear})$.

The complexity is about $\binom{n+r+1}{r+1}^w$, $(2 \le w < 3)$.

Smaller q^r is better for decryption,
larger r is better for security.

$\Rightarrow$ q should be small ($q = 2$ for most cases).

$\Rightarrow$ n should be large if q is small.

$\Rightarrow$ Key size is relatively large.

The original HFE is not very practical...

Arrangements.

Minus: Reducing (hiding) several quadratic forms.

Plus: Adding several quadratic forms.

Vinegar: Adding variables.

Projection: Reducing several variables.

Sflash (Patarin-Goubin-Courtois, 2001):
Minus of Matsumoto-Imai,
Selected by NESSIE, 2003,
Broken by the differencial attack (Dubois-Fouque-Shamir-Stern, 2007).

Quartz (Courtois et al., 2001), Gui (Petzold et al., 2015):
Minus and Vinegar of HFE (HFEv-).

Multi-HFE (Chen-Chen-Ding-Werner-Yang, 2008)

Constructed by Multivariate quadratic map.

$N > 1$,
$\mathcal{G} : K^N \to K^N$.

$$\mathcal{G}_l(X_1, \dots, X_N) = \sum_{1\le i\le j\le N} \alpha_{ij}^{(l)} X_i X_j + \sum_{1\le i\le N} \beta_i^{(l)} X_i + \gamma^{(l)}.$$

Decryption: Solve a system of N quadratic equations of N variables.

If N is small enough, it is efficient.

If N is large, a special structure of $\mathcal{G}$ is necessary to be inverted feasibley.

→ It can be used as "padding out" the size.

Security: Since

$$\mathcal{G}_l(X) = \tilde{X}^t \begin{pmatrix} *_N & \\ & \end{pmatrix} \tilde{X} + (\text{linear}),$$

the min-rank attack is available if N is small (Bettale-Faugere-Perret, 2013).
The complexity is about $\binom{n+N+1}{N+1}^w$, $(2 \le w < 3)$.

Since

$$f_l(x) = x^t \tilde{S}^t \begin{pmatrix} *_N & & \\ & \ddots & \\ & & *_N \end{pmatrix} \tilde{S} x + (\text{linear}),$$

the conjugation attack is available (H, 2015).
It is in polynmoial time (not highly depending on N).
⇒ "Padding out" is not useful.

HMFEv. (Petzoldt et al, 2017)

A vinegar variant of Multi-HFE.

The security against the min-rank attack and the conjugation attack is much better than Multi-HFE.

However, if N is small, the security against the high-rank attack is not enough (H, 2017).

ZHFE (Porras-Baena-Ding, 2014).

An over-defined type HFE using cubic polynomials in the decryption process.

$N = 1$, $M = 2$ $(m = 2n)$. $\mathcal{G} : K \to K^2$.
$\mathcal{G}_1(X), \mathcal{G}_2(X)$: quadratic forms of $(X, X^q, \dots, X^{q^{n-1}})$ s.t. the degree of

$$X \cdot \mathcal{G}_1(X) + X^q \cdot \mathcal{G}_2(X)$$

is small.

Decryption: For $Y_1 = \mathcal{G}_1(X)$, $Y_2 = \mathcal{G}_2(X)$, solve the univariate polynomial

$$X(\mathcal{G}_1(X) - Y_1) + X^q(\mathcal{G}_2(X) - Y_2) = 0.$$

Since the degree is small, it seems efficient.

Security: a little secure than HFE against the min-rank attack.
Using a CCA approach, one can recover the decryption polynoimal.
⇒ ZHFE is not much more practical than HFE.

Extension field type:

MI: a univariate monomial, broken.

HFE: a univariate polynomial, serious trade-off between security and efficiency, arrangements are better (?)

Multi-HFE: multivariate quadratic polynomials, broken.

ZHFE: two univariate polynomials, not much more secure than HFE.

Q. How about an extension "(non-commutative) ring", not an extension "field"?

A. Not recommended.

Artin-Wedderburn's theorem.

If the ring R is semi-simple, there exist integers $n_1, \ldots, n_l \geq 1$ and division rings $K_1, \ldots, K_l$ s.t.

$$R \simeq \mathrm{Mat}_{n_1}(K_1) \oplus \cdots \oplus \mathrm{Mat}_{n_l}(K_l).$$

→ Taking the basis carefully, the attacker can reduce the security against the rank attacks and the conjugation attacks.

Other schemes

ABC Encryption scheme. (Diene-Tao-Ding, 2013)

$r \geq 1$, $n = r^2$, $m = 2n$ (over-defined).

$A, B, C \in \mathrm{Mat}_r(k[x])$: entries are linear forms of $x_1, \ldots, x_n$.

$G_1 := AB$, $G_2 := AC$.

$G : k^n \to k^m$: the quadratic map given by the quadratic forms in G_1, G_2.

Decryption: Solve a system of linear equations of $x_1, \ldots, x_n$ derived from $B = C(G_2^{-1}G_1)$.

Security: $O(q^{\sqrt{n}} \cdot$ (polyn.) against the min-rank attack and the linearization attack.

The decryption fails with the probability (about) q^{-1}.

YTS signature scheme. (Yasuda-Takagi-Sakurai, 2013)

$r \geq 1$, $n = r^2$, $n \sim 2m$ (under-defined).

$X \in \mathrm{Mat}_r(k[x])$: the entries are $x_1, \ldots, x_n$.

$Y_1 := X^t X$, $Y_2 := X^t \begin{pmatrix} I_{r-1} & \\ & \delta \end{pmatrix} X$, $\quad (\delta/p) = -1$.

$G : k^n \to k^m$: the quadratic map given by the quadratic forms in Y_1, Y_2.

Signature generation: Find X such that $Y := X^t X$ or $Y := X^t \begin{pmatrix} I_{r-1} & \\ & \delta \end{pmatrix} X$.

Security: $O(q^{\sqrt{n}} \cdot$ (polyn.)) against the min-rank attack,
polynomial time by the conjugation attack. (H, 2014).

Summary and open problems

It seems that there are good **signature schemes** (UOV, Rainbow, HFEv-), while the key sizes are relatively large.

The **encryption schemes** seem less practical than the signature schemes.

It is (maybe) because generating a *one-to-one* quadratic map G is difficult.

If G is "strictly" one-to-one, G may have a special structure (then insecure?).

Q. The maps F, G are quadratic. How about (higher than) cubic?

The key size is larger.

The security is not much more than quadratic ones (weakness often appear in the quadratic parts).

Q. Are there security proofs?

A. Not at all. Most "secure" MPKCs are "presently" unbroken.

Q. Are there another expressions of MPKCs?

Writing down MPKCs over $\mathbb{F}_2$ by another (NP-complete or -hard) problems seems interesting.

On the Security of Homomorphic Encryption Schemes Based on Ring-LWE Problem over Decomposition Fields

Shinya Okumura (Osaka University)
This is a joint work with
Shota Terada, Hideto Nakano and Atsuko Miyaji (Osaka University)

1. Introduction

Motivation

- **Ring-LWE problem provides efficient cryptographic applications**
  - **Post-quantum public key cryptosystems**
  - **Fully homomorphic encryption (FHE) schemes**
  - **…**
- **Difficulty of Ring-LWE mainly depends on**
  - **Parameters on Noise**
  - **Underlying number fields**
  - **…**
- **Such schemes use Ring-LWE over cyclotomic fields from viewpoints of efficiency and security**

- (Especially, in the case of FHE) improving efficiency is still needed
- Arita and Handa proposed to use decomposition fields of cyclotomic fields

<u>Our Work</u>

- Experimental analysis of the security of Arita et al.'s HE scheme
  - Execute basic lattice attacks against Ring-LWE over
    - $\boldsymbol{m}$-th cyclotomic fields ($\boldsymbol{m}$: prime number)
    - Decomposition fields

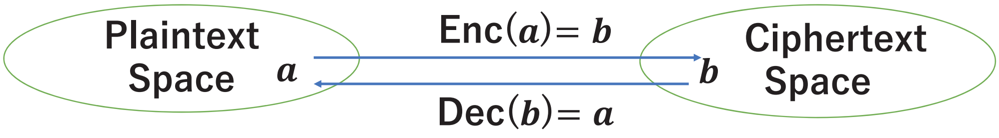
<u>Today's topics</u>

- Brief description of HE, Ring-LWE, Arita et al.'s idea, Lattice attack we used
- Show some our experimental results

2. Homomorphic Encryption

<u>Homomorphic Encryption Scheme</u>

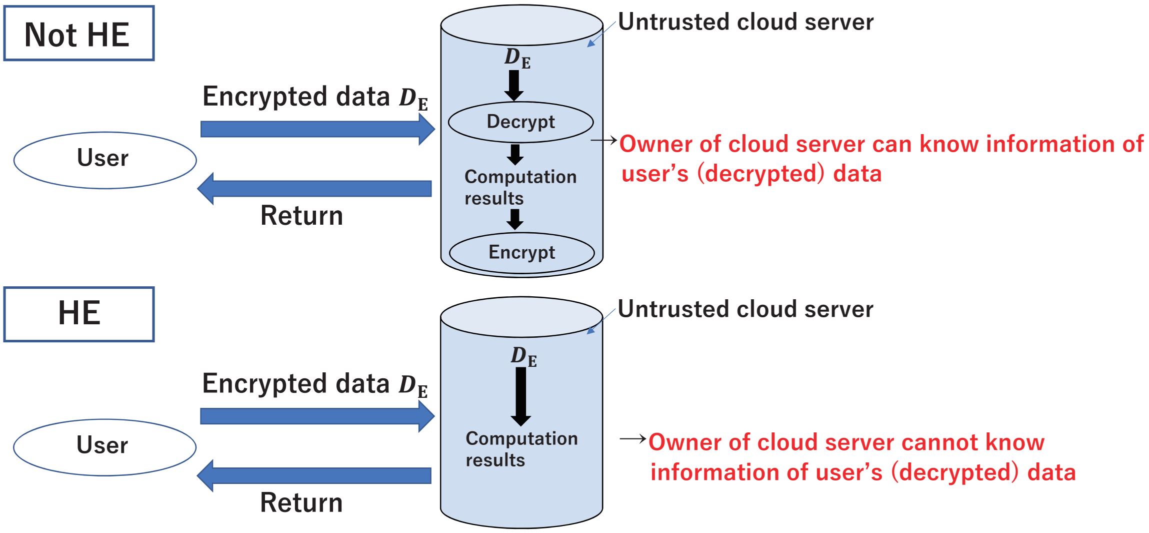
- Homomorphic encryption (HE) schemes can compute addition, multiplication or both operations of plaintexts without decrypting


Plaintext Space
$\boldsymbol{a}$
Enc($\boldsymbol{a}$)= $\boldsymbol{b}$
Dec($\boldsymbol{b}$)= $\boldsymbol{a}$
$\boldsymbol{b}$
Ciphertext Space

Additive HE : $\mathrm{Dec}(\mathrm{Enc}(\boldsymbol{a_1}) + (\times)\mathrm{Enc}(\boldsymbol{a_2})) = \boldsymbol{a_1} + \boldsymbol{a_2}$
Multiplicative HE : $\mathrm{Dec}(\mathrm{Enc}(\boldsymbol{a_1}) \times (+)\mathrm{Enc}(\boldsymbol{a_2})) = \boldsymbol{a_1} \times \boldsymbol{a_2}$
Additive+Multiplicative : Somewhat/Fully HE

Application

・HE schemes have many application to cloud computing area



Not HE

Encrypted data D_E

User

Return

Untrusted cloud server

D_E

Decrypt

Computation results

Encrypt

→Owner of cloud server can know information of user's (decrypted) data

HE

Encrypted data D_E

User

Return

Untrusted cloud server

D_E

Computation results

→Owner of cloud server cannot know information of user's (decrypted) data

3. Ring-LWE

Ring-LWE (RLWE) over number fields

K : Number field of $[K : \mathbb{Q}] = n$

$R = O_K$
$q \in \mathbb{Z}$
$R_q := R/qR$

$D_{\sigma,R,q}$: Discrete Gaussian distribution over R_q
(with mean 0 and variance σ^2)

$U(X)$:Uniform distribution over a set X

$a \hookleftarrow U(R_q),\ s \hookleftarrow U(R_q),\ e \hookleftarrow D_{\sigma,R,q},\ b = as + e$

(a, b) : RLWE sample

$\mathcal{R} := \mathcal{R}(R, q, \sigma, s)$: Set of RLWE samples

- Search Ring-LWE over K (SRLWE($\mathcal{R}$))
  Find s from arbitrary number of RLWE samples

$$(a_i, b_i = a_i s + e_i) \longleftarrow \mathcal{R}$$

- Decision Ring-LWE over K (DRLWE($\mathcal{R}$))
  Distinguish RLWE samples and samples from $U(R_q \times R_q)$

4. A Lattice Attack on Ring-LWE

Lattices

$n \in \mathbb{N}$

$\mathbb{b}_1, \dots, \mathbb{b}_m \in \mathbb{R}^n$: $\mathbb{R}$-linearly independent vectors $(m \leq n)$

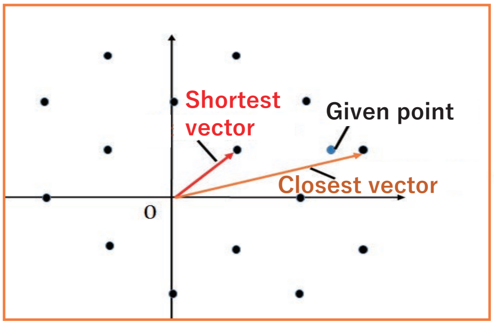
$\mathcal{L} := \mathbb{Z}\mathbb{b}_1 + \cdots + \mathbb{Z}\mathbb{b}_m$: Lattice in $\mathbb{R}^n$ with basis $\{\mathbb{b}_1, \dots, \mathbb{b}_m\}$

$\dim(\mathcal{L}) := n$: dimension of $\mathcal{L}$

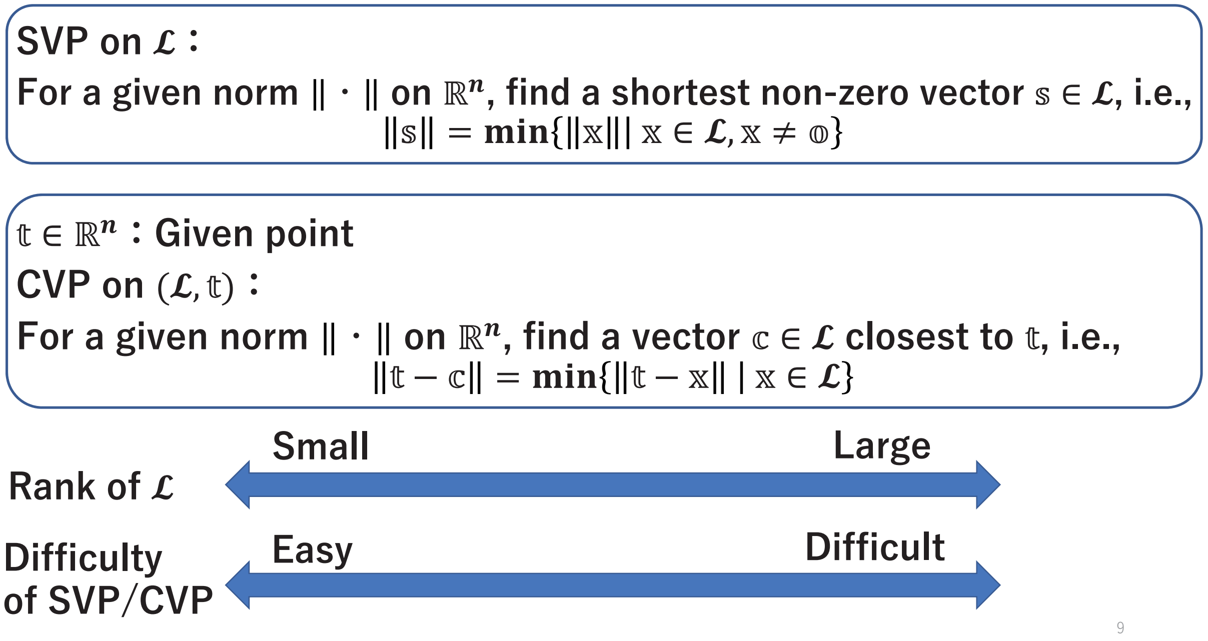
$\mathrm{rank}(\mathcal{L}) := m$: rank of $\mathcal{L}$

2. 1. 2. Problems on Lattices

1. Shortest Vector Problem (SVP)
2. Closest Vector Problem (CVP)



SVP and CVP on a 2-dimensional lattice



SVP on $\mathcal{L}$：

For a given norm ‖ · ‖ on $\mathbb{R}^n$, find a shortest non-zero vector $\mathbb{s} \in \mathcal{L}$, i.e.,

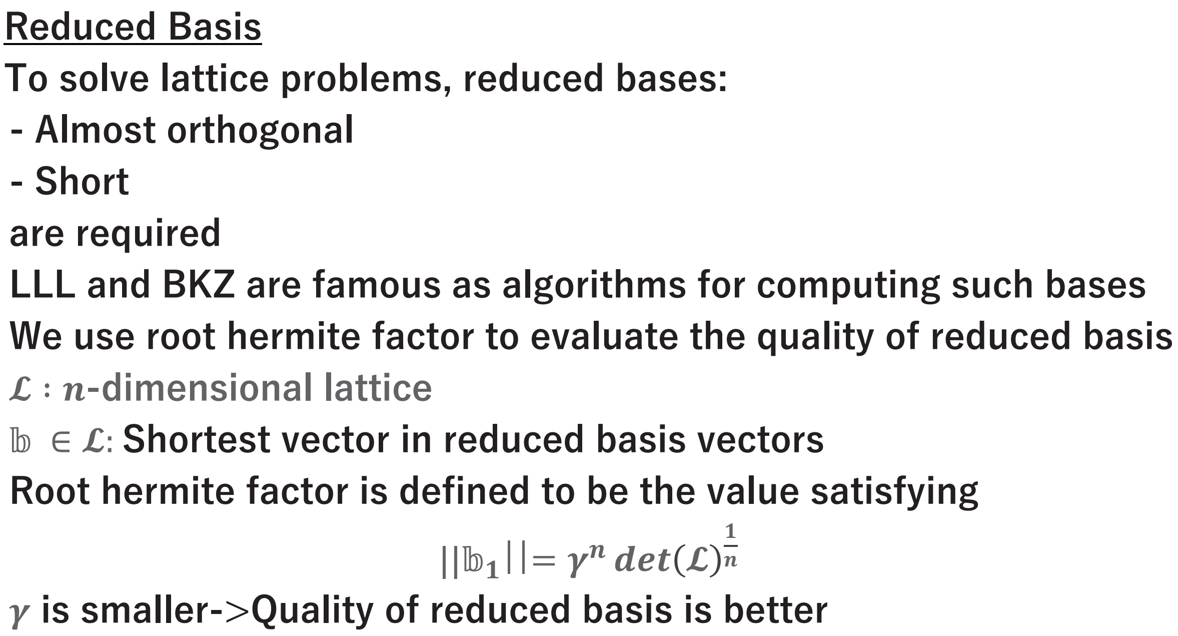
$$\|\mathbb{s}\| = \min\{\|\mathbb{x}\| | \, \mathbb{x} \in \mathcal{L}, \mathbb{x} \neq \mathbb{0}\}$$

$\mathbb{t} \in \mathbb{R}^n$：Given point

CVP on $(\mathcal{L}, \mathbb{t})$：

For a given norm ‖ · ‖ on $\mathbb{R}^n$, find a vector $\mathbb{c} \in \mathcal{L}$ closest to $\mathbb{t}$, i.e.,

$$\|\mathbb{t} - \mathbb{c}\| = \min\{\|\mathbb{t} - \mathbb{x}\| \mid \mathbb{x} \in \mathcal{L}\}$$



Reduced Basis

To solve lattice problems, reduced bases:

- Almost orthogonal
- Short

are required

LLL and BKZ are famous as algorithms for computing such bases

We use root hermite factor to evaluate the quality of reduced basis

$\mathcal{L}$: n-dimensional lattice

$\mathbb{b} \in \mathcal{L}$: Shortest vector in reduced basis vectors

Root hermite factor is defined to be the value satisfying

$$||\mathbb{b}_1|| = \gamma^n \, det(\mathcal{L})^{\frac{1}{n}}$$

γ is smaller->Quality of reduced basis is better

Lattice Attack [1]

$\boldsymbol{\mu_0, \mu_1, \dots, \mu_{n-1}}$: $\mathbb{Z}$-basis of $\boldsymbol{R}$

$(\boldsymbol{a_0, a_1, \dots, a_{n-1}})^T \in (\mathbb{Z}/q\mathbb{Z})^n$: Uniformly random

$(\boldsymbol{s_0, s_1, \dots, s_{n-1}})^T \in (\mathbb{Z}/q\mathbb{Z})^n$: Random ($|s_i| \leq 1$ in the case of FHE)

$(\boldsymbol{e_0, e_1, \dots, e_{n-1}})^T \in (\mathbb{Z}/q\mathbb{Z})^n$: From $\boldsymbol{D_{\sigma,R,q}}$ ($q \gg \sigma$)

$\boldsymbol{a = a_0\mu_0 + a_1\mu_1 + \cdots + a_{n-1}\mu_{n-1}}$

$\boldsymbol{s = s_0\mu_0 + s_1\mu_1 + \cdots + s_{n-1}\mu_{n-1}}$

$\boldsymbol{e = e_0\mu_0 + e_1\mu_1 + \cdots + e_{n-1}\mu_{n-1}}$

$\boldsymbol{b = as + e = b_0\mu_0 + b_1\mu_1 + \cdots + b_{n-1}\mu_{n-1}}$

$\mathbb{b} = (\boldsymbol{b_0, b_1, \dots, b_{n-1}})^T, \mathbb{s} = (\boldsymbol{s'_0, s'_1, \dots, s'_{n-1}})^T, \mathbb{e} = (\boldsymbol{e'_0, e'_1, \dots, e'_{n-1}})^T$

$\boldsymbol{s'_i, e'_i}$: Variables

[2] Guillaume Bonnoron, Caroline Fontaine "A Note on Ring-LWE Security in the Case of Fully Homomorphic Encryption", INDOCRYPT 2017, LNCS, vol. 10698, pp. 27-43, Springer, Cham, 2017.

$\boldsymbol{b = as + e}$

$\Longrightarrow (\boldsymbol{s_0, s_1, \dots, s_{n-1}, e_0, e_1, \dots, e_{n-1}})^T$ is a (short) solution to

$\boldsymbol{A}\mathbb{s} + \mathbb{e} = \mathbb{b}$ (mod. q)$\cdots$($*$) $\left({}^{\exists}\boldsymbol{A} \in \boldsymbol{M_n}(\mathbb{Z}/q\mathbb{Z})\right)$

Short solution to ($*$) can be expected to be

$(\boldsymbol{s_0, s_1, \dots, s_{n-1}, e_0, e_1, \dots, e_{n-1}})^T$

・Method of finding short solution

$\boldsymbol{A'} \coloneqq (\boldsymbol{A}\ \ \boldsymbol{I})$,

$(*) \Longrightarrow \boldsymbol{A_3}\binom{\mathbb{s}}{\mathbb{e}} = \mathbb{b}$ (mod. q)$\cdots$(※)

$\binom{\mathbb{O}}{\mathbb{b}}$: Solution to (※)

$\boldsymbol{\mathcal{L}'} \coloneqq \left\{\mathbb{v} \in \mathbb{Z}^{2n} \middle| \boldsymbol{A'}\mathbb{v} = \mathbb{O} \text{ (mod. } q)\right\}$: Lattice

column vectors of $\begin{pmatrix} \boldsymbol{I} & \mathbb{O} \\ -\boldsymbol{A} & \boldsymbol{qI} \end{pmatrix}$ form a basis of $\boldsymbol{\mathcal{L}'}$

$\mathbb{f}$: Solution to approximate CVP w.r.t. $\mathcal{L}_3$ and $\binom{\mathbb{0}}{\mathbb{b}}$

$\binom{\tilde{\mathbb{s}}}{\tilde{\mathbb{e}}} \coloneqq \binom{\mathbb{0}}{\mathbb{b}} - \mathbb{f}$

Each entry of $\binom{\tilde{\mathbb{s}}}{\tilde{\mathbb{e}}}$ is small $\Rightarrow (\boldsymbol{a}, \boldsymbol{b})$: (Probably) RLWE sample

We used
- Babai's nearest plane algorithm
- Kannan's embedding technique

to solve approximate CVP

5. Arita-Handa's Propopsal

Plaintext slots of HE based on Ring-LWE

$\boldsymbol{\zeta_m}$: $\boldsymbol{m}$-th root of unity ($\mathbb{N} \ni \boldsymbol{m} > \mathbf{2}$, prime)

$\boldsymbol{K} \coloneqq \mathbb{Q}(\boldsymbol{\zeta_m})$: $\boldsymbol{m}$-th cyclotomic field

$\boldsymbol{\varphi}(\boldsymbol{m}) = |(\mathbb{Z}/\boldsymbol{m}\mathbb{Z})^{\times}|$

$\boldsymbol{R} \coloneqq \mathbb{Z}[\boldsymbol{\zeta_m}]$: ring of integers of $\boldsymbol{K}$

$\boldsymbol{p}$: prime number

$\boldsymbol{pR}$: ideal of $\boldsymbol{R}$ generated by $\boldsymbol{p}$

$\boldsymbol{p} \nmid \boldsymbol{m} \Rightarrow \boldsymbol{pR} = \mathfrak{B}_0\mathfrak{B}_1 \cdots \mathfrak{B}_{g-1}$: prime ideal decomposition

($\mathfrak{B}_i$: prime ideal, $\boldsymbol{i} \neq \boldsymbol{j} \Rightarrow \mathfrak{B}_i \neq \mathfrak{B}_j$)

$\boldsymbol{R}/\mathfrak{B}_i \cong \mathbb{F}_{p^d}$ $(\boldsymbol{d} = \boldsymbol{\varphi}(\boldsymbol{m})/\boldsymbol{g})$

R/pR : plaintext space

$R/pR \cong \underbrace{\mathbb{F}_{p^d} \oplus \cdots \oplus \mathbb{F}_{p^d}}_{g \text{ plaintext slots}} \cdots (*)$ (Chinise Remainder Theorem)

g plaintext slots ➡ g plaintexts can be encrypted at the same time

$(*)$ ➡HE schemes need arithmetics on $\mathbb{F}_{p^d}$

d may be large ➡HE will be inefficient

※$p \equiv 1$ (mod. m)$\Rightarrow p$ splits completely in $K \Rightarrow d = 1$

However, p should be small from the viewpoint of efficiency

E.g. $p = 2$

Arita et al.'s idea [2]

Use a subring R_Z of R (called the decomposition ring w.r.t. p) having special properties :

(i) $R_Z/pR_Z \cong \mathbb{F}_p \oplus \cdots \oplus \mathbb{F}_p$

・Rank of lattices occurring in lattice attack is

- mth cyclotomic field : $m-1$
- Decomposition field : g

・The number of plaintext slots

- mth cyclotomic field : $\frac{m-1}{d}$
- Decomposition field : g

[2] Seiko Arita and Sari Handa "Subring Homomorphic Encryption", accepted to ICISC 2017.

(ii) Advantages of R (for cryptographic application) are inherited

- Having good bases
- Equivalence of search Ring-LWE and decision Ring-LWE
- Quantum polynomial-time reduction from approximate shortest vector problem on ideal lattices exists

Arita et al. constructed a HE scheme with IND-CPA secure
if Ring-LWE over R_Z is hard
※Arita et al. assume m is a prime number

<u>Decomposition Field and Decomposition Ring</u>

$G = \mathrm{Gal}(K/\mathbb{Q}) \coloneqq \{\sigma: K \cong K | \sigma(a) = a, \forall a \in \mathbb{Q}\}$

$G_{\mathfrak{B}} \coloneqq \{\sigma \in G \mid \sigma(\mathfrak{B}_i) = \mathfrak{B}_i\ (i = 0, \dots, g-1)\}$: Decomposition group of K w.r.t. p

$Z \coloneqq \{a \in K | \sigma(a) = a, \forall \sigma \in G_{\mathfrak{B}}\}$: Decomposition field of K w.r.t. p

$R_Z \coloneqq R \cap Z$: Decomposition Ring

$\mathfrak{p}_i \coloneqq R_Z \cap \mathfrak{B}_i$

$G_{\mathfrak{B}}$ acts on $R/\mathfrak{B}_i \cong \mathbb{F}_{p^d}$ as pth Frobenius map

i.e., $\sigma(x) \equiv x^p \pmod{\mathfrak{B}_i}$, $\forall \sigma \in G_{\mathfrak{B}}$, $\forall x \in R$

➡ $R_Z/\mathfrak{p}_i \cong \mathbb{F}_p$ ➡ $R_Z/pR_Z \cong \underbrace{\mathbb{F}_p \oplus \cdots \oplus \mathbb{F}_p}_{g \text{ plaintext slots}}$: plaintext space

$q = p^{\ell}$

$R_Z/qR_Z \cong \mathbb{Z}/p^{\ell}\mathbb{Z} \oplus \cdots \oplus \mathbb{Z}/p^{\ell}\mathbb{Z}$: plaintext space in some application

Gal$(K/\mathbb{Q}) \cong (\mathbb{Z}/m\mathbb{Z})^*$

$G_{\mathfrak{B}} \cong \langle p$ (mod. $m)\rangle \subset (\mathbb{Z}/m\mathbb{Z})^*$

$\{\bar{t}_0, \dots, \bar{t}_{g-1}\}$: Complete set of representatives of $(\mathbb{Z}/m\mathbb{Z})^*/\langle p\rangle$

$t_0, \dots, t_{g-1} \in \mathbb{Z}$ such that t_i represents $\bar{t}_i$ for all $0 \le i \le g-1$

$d = |G_{\mathfrak{B}}|$

$$\eta_i := \sum_{0 \le k \le d-1} {\zeta_m}^{p^k t_i} \quad (0 \le i \le g-1)$$

$\eta_0, \dots, \eta_{g-1}$ is a $\mathbb{Z}$-basis of R_Z

6. Our Experiments

Procedure of Our Experiments

Parameters：$p = 2, q = 2^r, r' < r, q' = 2^{r'}$

1. Generate 100 RLWE samples (mod. q)
2. Execute samples (mod. q')
2. Construct lattices from samples
3. Execute Two attacks against search Ring-LWE
   - Babai’s nearest plane algorithm
     (i) Apply BKZ with block size = 10 to $\begin{pmatrix} I & \mathbb{O} \\ -A & qI \end{pmatrix}$
     (ii) Compute root hermite factor to evaluate the quality of BKZ
     (iii) Apply Babai’s nearest plane algorithm to BKZ reduced basis

- Kannan's embedding technique
  - (i) Construct lattices $\begin{pmatrix} I & \mathbb{O} & t \\ -A & qI & \\ & \mathbb{O} & 1 \end{pmatrix}$ from $\begin{pmatrix} I & \mathbb{O} \\ -A & qI \end{pmatrix}$ and $t = \begin{pmatrix} \mathbb{O} \\ \mathbb{b} \end{pmatrix}$
  - (ii) Apply BKZ to $\begin{pmatrix} I & \mathbb{O} & t \\ -A & qI & \\ & \mathbb{O} & 1 \end{pmatrix}$ (After that compute root hermite factor)

We used the following CPU and software

- CPU: Intel(R) Xeon(R) CPU E7-4830 v4 (2.00GHz) × 4
- RAM: 1534GB
- OS: Ubuntu 16.04
- SageMath version 7.5.1(Sample generation)
- Magma version 2.23-1(Attack)

Experimental Results (Babai's nearest plane algorithm)

Cyclotomic Field			
m	73	83	107
r	180	180	180
r'	20	20	20
The number of successes/100	100/ 100	100/ 100	20/ 100
Average of root hermite factors	1.0144	1.0144	1.0196
Average of running times [sec]	261.22	520.171	2402.339

Decomposition Field			
m	1801	4051	2731
g	72	82	105
r	180	180	180
r'	20	20	20
The number of successes/100	100/ 100	100/ 100	33/ 100
Average of root hermite factors	1.0144	1.0145	1.0196
Average running times [sec]	238.419	480.640	2512.291

Experimental Results (Kannan's embedding technique)

Cyclotomic Field			
m	73	83	107
r	180	180	180
r'	20	20	20
The number of successes/100	100/ 100	100/ 100	74/ 100
Average of root hermite factors	0.9818	0.9843	0.9966
Average of running times [sec]	37.671	95.961	540.751

Decomposition Field			
m	1801	4051	2731
g	72	82	105
r	180	180	180
r'	20	20	20
The number of successes/100	100/ 100	100/ 100	58/ 74 *
Average of root hermite factors	0.9818	0.9841	0.9952
Average running times [sec]	38.880	96.865	769.902

* BKZ did not terminate, and so we give incomplete result

7. Conclusion and Future Work

Conclusion

1. We executed (basic) lattice attacks
   - Babai's nearest plane algorithm
   - Kannan's embedding technique

   against Ring-LWE over decomposition/cyclotomic fields
2. We could not find the disadvantage of using Ring-LWE over decomposition fields, but more analysis should be done

Future Work

- Execute more experiments on lattice/other attacks
- Improve attacks or find advantage by using properties of decomposition fields

Thank you for your attention!

「マス・フォア・インダストリ研究」シリーズ刊行にあたり

本シリーズは，平成 23 年 4 月に設立された九州大学マス・フォア・ インダストリ研究所(IMI)が，平成 25 年 4 月に共同利用・共同研究拠点「産業数学の先進的・基礎的共同研究拠点」として，文部科学大臣より認定を受けたことにともない刊行するものである．本シリーズでは，主として，マス・フォア・インダストリに関する研究集会の会議録，共同研究の成果報告等を出版する．　各巻はマス・フォア・インダストリの最新の研究成果に加え，その新たな視点からのサーベイ及びレビューなども収録し，マス・フォア・インダストリの展開に資するものとする．

平成 26 年 10 月
マス・フォア・インダストリ研究所
所長　福本康秀

代数的手法による数理暗号解析

マス・フォア・インダストリ研究　No.9, IMI,　九州大学

ISSN 2188-286X

発行日　　2018 年 3 月 1 日

編　集　　阿部拓郎, 高島克幸, 縫田光司, 安田雅哉

発　行　　九州大学マス・フォア・インダストリ研究所
〒819-0395 福岡市西区元岡 744
九州大学数理・IMI 事務室
TEL 092-802-4402　FAX 092-802-4405
URL http://www.imi.kyushu-u.ac.jp/

印　刷　　社会福祉法人　福岡コロニー
〒811-0119　福岡県糟屋郡新宮町緑ケ浜 1 丁目 11 番 1 号
TEL 092-962-0764　FAX 092-962-0768

シリーズ既刊

Issue	Author ／ Editor	Title	Published
マス・フォア・インダストリ 研究　No.1	穴田 啓晃 安田 貴徳 Xavier Dahan 櫻井 幸一	Functional Encryption as a Social Infrastructure and Its Realization by Elliptic Curves and Lattices	26 February 2015
マス・フォア・インダストリ 研究　No.2	滝口 孝志 藤原 宏志	Collaboration Between Theory and Practice in Inverse Problems	12 March 2015
マス・フォア・インダストリ 研究　No.3	筧 三郎	非線形数理モデルの諸相：連続，離散，超離散，その先 (Various aspects of nonlinear mathematical models : continuous, discrete, ultra-discrete, and beyond)	24 March 2015
マス・フォア・インダストリ 研究 No.4	穴田　啓晃 安田　貴徳 櫻井　幸一 寺西　勇	Next-generation Cryptography for Privacy Protection and Decentralized Control and Mathematical Structures to Support Techniques	29 January 2016
マス・フォア・インダストリ 研究 No.5	藤原　宏志 滝口　孝志	Mathematical Backgrounds and Future Progress of Practical Inverse Problems	1 March 2016
マス・フォア・インダストリ 研究 No.6	松谷　茂樹 佐伯　修 中川　淳一 上坂　正晃 濱田　裕康	結晶のらせん転位の数理	10 January 2017
マス・フォア・インダストリ 研究 No.7	滝口　孝志 藤原　宏志	Collaboration among mathematics, engineering and industry on various problems in infrastructure and environment	1 March 2017
マス・フォア・インダストリ 研究 No.8	藤原　宏志 滝口　孝志	Practical inverse problems based on interdisciplinary and industry-academia collaboration	20 February 2018

Institute of Mathematics for Industry
Kyushu University

九州大学マス・フォア・インダストリ研究所

〒819-0395 福岡市西区元岡744
URL http://www.imi.kyushu-u.ac.jp/