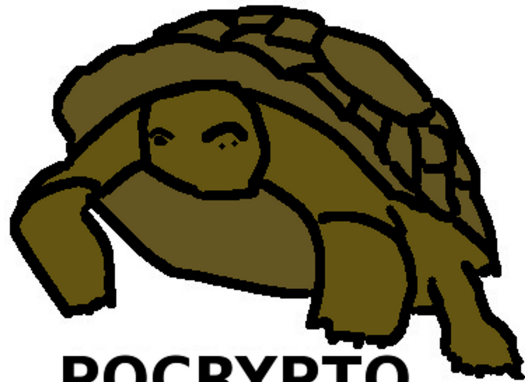
Post-quantum cryptography

Tanja Lange



23 March 2016

BeNeLux Mathematical Congress

Cryptography

- Motivation #1: Communication channels are spying on our data.
- Motivation #2: Communication channels are modifying our data.

- Literal meaning of cryptography: "secret writing".
- Achieves various security goals by secretly transforming messages.

iacrmemHEREATiacr.org

INTERNATIONAL ASSOCIATION FOR CRYPTOLOGIC RESEARCH

2013

1702 Members
(1580 in 2012)

1245 Regular+

457 Students


https://www.iacr.org/
https://www.iacr.org/docs/minutes/c2013mem-slides.pdf
IACR Mer
16 / 55
www.iacr.org
Your connection to this site is private.
Permissions
Connection
The identity of this website has been verified by RapidSSL SHA256 CA - G3.
No Certificate Transparency information was supplied by the server.
Certificate information
Your connection to www.iacr.org is encrypted using a modern cipher suite.
The connection uses TLS 1.2.
The connection is encrypted and authenticated using AES_128_GCM and uses ECDHE_RSA as the key exchange mechanism.
What do these mean?
iacrmemH
INTERNATIONAL ASSOCIATION FOR CRYPTOLOGIC RESEARCH
2013

Cryptographic applications in daily life

- Mobile phones connecting to cell towers.
- Credit cards, EC-cards, access codes for Rabobank.
- Electronic passports; soon ID cards.
- Internet commerce, online tax declarations, webmail.
- Any webpage with `https`.
- Encrypted file system on iPhone (see Apple vs. FBI).
- Facebook, WhatsApp, iMessage on iPhone.

Cryptographic applications in daily life

- Mobile phones connecting to cell towers.
- Credit cards, EC-cards, access codes for Rabobank.
- Electronic passports; soon ID cards.
- Internet commerce, online tax declarations, webmail.
- Any webpage with `https`.
- Encrypted file system on iPhone (see Apple vs. FBI).
- Facebook, WhatsApp, iMessage on iPhone.
- PGP encrypted email, Signal, Tor, Tails Qubes OS

Cryptographic applications in daily life

- Mobile phones connecting to cell towers.
- Credit cards, EC-cards, access codes for Rabobank.
- Electronic passports; soon ID cards.
- Internet commerce, online tax declarations, webmail.
- Any webpage with `https`.
- Encrypted file system on iPhone (see Apple vs. FBI).
- Facebook, WhatsApp, iMessage on iPhone.
- PGP encrypted email, Signal, Tor, Tails Qubes OS

Snowden in Reddit AmA

> *Arguing that you don't care about the right to privacy because you have nothing to hide is no different than saying you don't care about free speech because you have nothing to say.*

Cryptographic tools

Many factors influence the security and privacy of data

- Secure storage, physical security; access control.
- Protection against alteration of data
  ⇒ digital signatures, message authentication codes.
- Protection of sensitive content against reading
  ⇒ encryption.

Cryptology is the science that studies mathematical techniques in order to provide secrecy, authenticity and related properties for digital information.

Currently used crypto (check the lock icon in your browser) starts with RSA, Diffie-Hellman (DH) in finite fields, or elliptic curve DH followed by AES or ChaCha20.
Newer systems: Curve25519, and Ed25519.

Security is getting better, but lots of bugs and no secure hardware

Cryptographic tools

Many factors influence the security and privacy of data

- Secure storage, physical security; access control.
- Protection against alteration of data
  ⇒ digital signatures, message authentication codes.
- Protection of sensitive content against reading
  ⇒ encryption.

Cryptology is the science that studies mathematical techniques in order to provide secrecy, authenticity and related properties for digital information.

Currently used crypto (check the lock icon in your browser) starts with RSA, Diffie-Hellman (DH) in finite fields, or elliptic curve DH followed by AES or ChaCha20.
Newer systems: Curve25519, and Ed25519.

Security is getting better, but lots of bugs and no secure hardware – let alone anti-security measures such as the Dutch "Hackvoorstel".

Algorithms for Quantum Computation: Discrete Logarithms and Factoring

Peter W. Shor
AT&T Bell Labs
Room 2D-149
600 Mountain Ave.
Murray Hill, NJ 07974, USA

Abstract

A computer is generally considered to be a universal computational device; i.e., it is believed able to simulate any physical computational device with a cost in computation time of at most a polynomial factor. It is not clear whether this is still true when quantum mechanics is taken into consideration. Several researchers, starting with David Deutsch, have developed models for quantum mechanical computers and have investigated their computational properties. This paper gives Las Vegas algorithms for finding discrete logarithms and factoring integers on a quantum computer that take a number of steps which is polynomial in the input size, e.g., the number of digits of the integer to be factored. These two problems are generally considered hard on a classical computer and have been used as the basis of several proposed cryptosystems. (We

[1, 2]. Although he did not ask whether quantum mechanics conferred extra power to computation, he did show that a Turing machine could be simulated by the reversible unitary evolution of a quantum process, which is a necessary prerequisite for quantum computation. Deutsch [9, 10] was the first to give an explicit model of quantum computation. He defined both quantum Turing machines and quantum circuits and investigated some of their properties.

The next part of this paper discusses how quantum computation relates to classical complexity classes. We will thus first give a brief intuitive discussion of complexity classes for those readers who do not have this background. There are generally two resources which limit the ability of computers to solve large problems: time and space (i.e., memory). The field of analysis of algorithms considers the asymptotic demands that algorithms make for these resources as a function of the problem size. Theoretical

In the long term, all encryption needs to be post-quantum

- Mark Ketchen, IBM Research, 2012, on quantum computing: "Were actually doing things that are making us think like, 'hey this isn't 50 years off, this is maybe just 10 years off, or 15 years off.' It's within reach."

In the long term, all encryption needs to be post-quantum

- Mark Ketchen, IBM Research, 2012, on quantum computing: "Were actually doing things that are making us think like, 'hey this isn't 50 years off, this is maybe just 10 years off, or 15 years off.' It's within reach."
- Fast-forward to 2022, or 2027. Quantum computers exist.
- Shor's algorithm solves in polynomial time:
  - Integer factorization.
  - The discrete-logarithm problem in finite fields.
  - The discrete-logarithm problem on elliptic curves.
- This breaks all current public-key encryption on the Internet!

In the long term, all encryption needs to be post-quantum

- Mark Ketchen, IBM Research, 2012, on quantum computing: "Were actually doing things that are making us think like, 'hey this isn't 50 years off, this is maybe just 10 years off, or 15 years off.' It's within reach."
- Fast-forward to 2022, or 2027. Quantum computers exist.
- Shor's algorithm solves in polynomial time:
  - Integer factorization.
  - The discrete-logarithm problem in finite fields.
  - The discrete-logarithm problem on elliptic curves.
- This breaks all current public-key encryption on the Internet!
- Also, Grover's algorithm speeds up brute-force searches.
- Example: Only 2^{64} quantum operations to break AES-128.

In the long term, all encryption needs to be post-quantum

- Mark Ketchen, IBM Research, 2012, on quantum computing: "Were actually doing things that are making us think like, 'hey this isn't 50 years off, this is maybe just 10 years off, or 15 years off.' It's within reach."
- Fast-forward to 2022, or 2027. Quantum computers exist.
- Shor's algorithm solves in polynomial time:
  - Integer factorization.
  - The discrete-logarithm problem in finite fields.
  - The discrete-logarithm problem on elliptic curves.
- This breaks all current public-key encryption on the Internet!
- Also, Grover's algorithm speeds up brute-force searches.
- Example: Only 2^{64} quantum operations to break AES-128.
- Need to switch the Internet to post-quantum encryption.

Confidence-inspiring crypto takes time to build

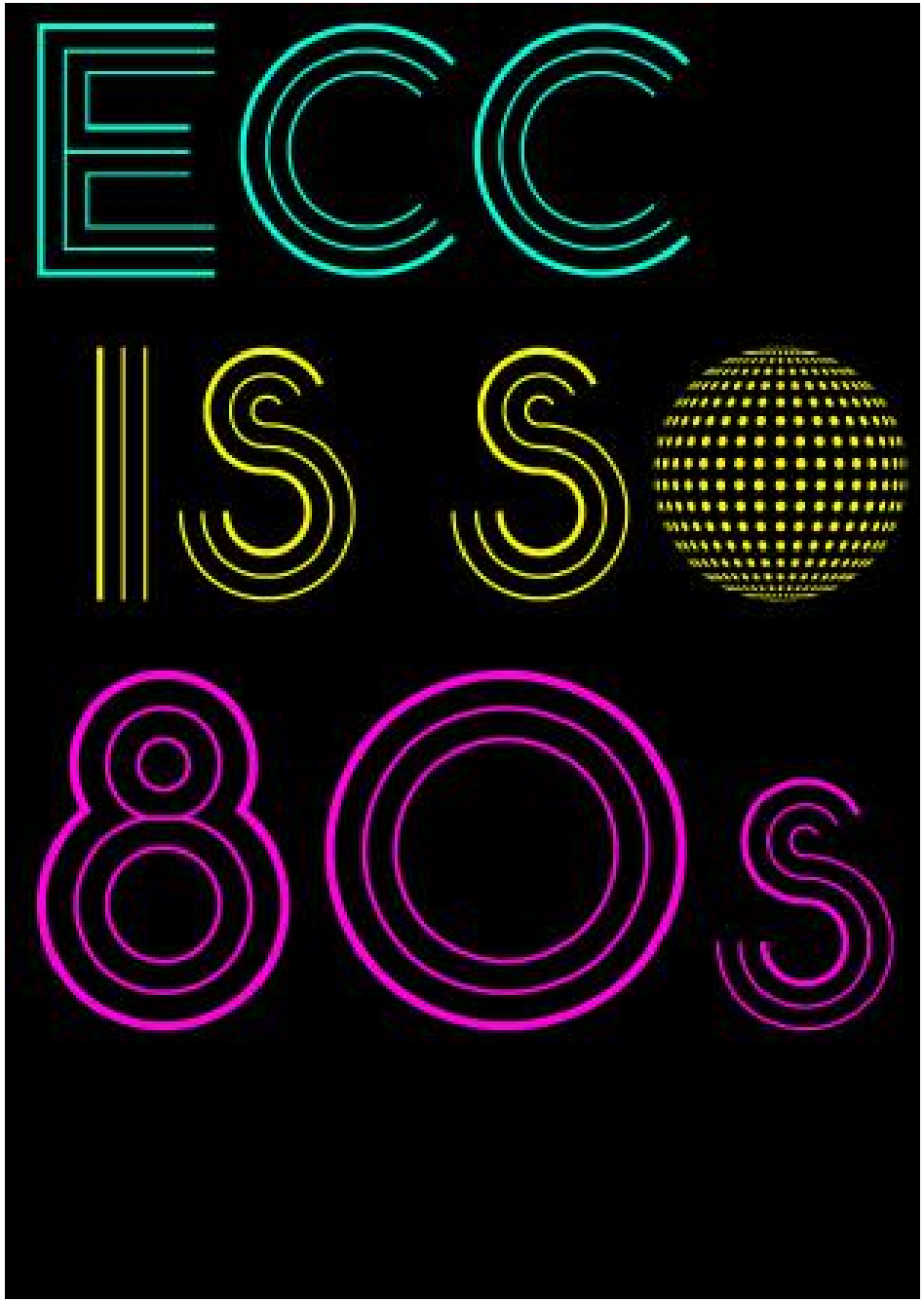
- Many stages of research from cryptographic design to deployment:
  - Explore space of cryptosystems.
  - Study algorithms for the attackers.
  - Focus on secure cryptosystems.

Confidence-inspiring crypto takes time to build

- Many stages of research from cryptographic design to deployment:
  - Explore space of cryptosystems.
  - Study algorithms for the attackers.
  - Focus on secure cryptosystems.
  - Study algorithms for the users.
  - Study implementations on real hardware.
  - Study side-channel attacks, fault attacks, etc.
  - Focus on secure, reliable implementations.
  - Focus on implementations meeting performance requirements.
  - Integrate securely into real-world applications.

Confidence-inspiring crypto takes time to build

- Many stages of research from cryptographic design to deployment:
  - Explore space of cryptosystems.
  - Study algorithms for the attackers.
  - Focus on secure cryptosystems.
  - Study algorithms for the users.
  - Study implementations on real hardware.
  - Study side-channel attacks, fault attacks, etc.
  - Focus on secure, reliable implementations.
  - Focus on implementations meeting performance requirements.
  - Integrate securely into real-world applications.
- Example: ECC introduced **1985**; big advantages over RSA. Robust ECC is starting to take over the Internet in **2015**.
- Post-quantum research can't wait for quantum computers!


ECC
IS
80s

Even higher urgency for long-term confidentiality

- Today's encrypted communication is being stored by attackers and will be decrypted years later with quantum computers. Danger for human-rights workers, medical records, journalists, security research, legal proceedings, state secrets, ...

Impact of PQCRYPTO (EU project in Horizon 2020)

- All currently used public-key systems on the Internet are broken by quantum computers.
- Today's encrypted communication can be (and is being!) stored by attackers and can be decrypted later with quantum computer.
- Post-quantum secure cryptosystems exist but are under-researched – we can recommend secure systems now, but they are big and slow

Impact of PQCRYPTO (EU project in Horizon 2020)

- All currently used public-key systems on the Internet are broken by quantum computers.
- Today's encrypted communication can be (and is being!) stored by attackers and can be decrypted later with quantum computer.
- Post-quantum secure cryptosystems exist but are under-researched – we can recommend secure systems now, but they are big and slow hence the logo.

Impact of PQCRYPTO (EU project in Horizon 2020)

- All currently used public-key systems on the Internet are broken by quantum computers.
- Today's encrypted communication can be (and is being!) stored by attackers and can be decrypted later with quantum computer.
- Post-quantum secure cryptosystems exist but are under-researched – we can recommend secure systems now, but they are big and slow hence the logo.
- PQCRYPTO will design a portfolio of high-security post-quantum public-key systems, and will improve the speed of these systems, adapting to the different performance challenges of mobile devices, the cloud, and the Internet.
- PQCRYPTO will provide efficient implementations of high-security post-quantum cryptography for a broad spectrum of real-world applications.

Initial recommendations of long-term secure post-quantum systems

Daniel Augot, Lejla Batina, Daniel J. Bernstein, Joppe Bos, Johannes Buchmann, Wouter Castryck, Orr Dunkelman, Tim Güneysu, Shay Gueron, Andreas Hülsing, Tanja Lange, Mohamed Saied Emam Mohamed, Christian Rechberger, Peter Schwabe, Nicolas Sendrier, Frederik Vercauteren, Bo-Yin Yang

Initial recommendations

- **Symmetric encryption** Thoroughly analyzed, 256-bit keys:
  - AES-256
  - Salsa20 with a 256-bit key

  Evaluating: Serpent-256, ...
- **Symmetric authentication** Information-theoretic MACs:
  - GCM using a 96-bit nonce and a 128-bit authenticator
  - Poly1305
- **Public-key encryption** McEliece with binary Goppa codes:
  - length $n = 6960$, dimension $k = 5413$, $t = 119$ errors

  Evaluating: QC-MDPC, Stehlé-Steinfeld NTRU, ...
- **Public-key signatures** Hash-based (minimal assumptions):
  - XMSS with any of the parameters specified in CFRG draft
  - SPHINCS-256

  Evaluating: HFEv-, ...

Hamming code

Parity check matrix ($n = 7, k = 4$):

$$H = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}$$

An error-free string of 7 bits $\mathbf{b} = (b_0, b_1, b_2, b_3, b_4, b_5, b_6)$ satisfies these three equations:

$$\begin{array}{ccccccccccc} b_0 & +b_1 & & & +b_3 & +b_4 & & & & = & 0 \\ b_0 & & & +b_2 & +b_3 & & & +b_5 & & = & 0 \\ & & b_1 & +b_2 & +b_3 & & & & +b_6 & = & 0 \end{array}$$

If one error occurred at least one of these equations will not hold.
Failure pattern uniquely identifies the error location,
e.g., $1, 0, 1$ means

Hamming code

Parity check matrix $(n = 7, k = 4)$:

$$H = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}$$

An error-free string of 7 bits $\mathbf{b} = (b_0, b_1, b_2, b_3, b_4, b_5, b_6)$ satisfies these three equations:

$$\begin{array}{ccccccccccc} b_0 & +b_1 & & & +b_3 & +b_4 & & & & = & 0 \\ b_0 & & & +b_2 & +b_3 & & & +b_5 & & = & 0 \\ & & b_1 & +b_2 & +b_3 & & & & +b_6 & = & 0 \end{array}$$

If one error occurred at least one of these equations will not hold.
Failure pattern uniquely identifies the error location,
e.g., $1, 0, 1$ means b_1 flipped.

Hamming code

Parity check matrix $(n = 7, k = 4)$:

$$H = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}$$

An error-free string of 7 bits $\mathbf{b} = (b_0, b_1, b_2, b_3, b_4, b_5, b_6)$ satisfies these three equations:

$$\begin{array}{cccccccccc} b_0 & +b_1 & & & +b_3 & +b_4 & & & = & 0 \\ b_0 & & +b_2 & +b_3 & & & +b_5 & & = & 0 \\ & b_1 & +b_2 & +b_3 & & & & +b_6 & = & 0 \end{array}$$

If one error occurred at least one of these equations will not hold.
Failure pattern uniquely identifies the error location,
e.g., $1, 0, 1$ means b_1 flipped.
The failure pattern $H \cdot \mathbf{b}$ is called the syndrome.

Coding theory

- Names: code word $\mathbf{c}$, error vector $\mathbf{e}$, received word $\mathbf{b} = \mathbf{c} + \mathbf{e}$.
- Very common to transform the matrix to have identity matrix on the right (no need to store that).

$$H = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix} \rightsquigarrow \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}$$

- Many special constructions discovered in 65 years of coding theory:
  - Large matrix H.
  - Fast decoding algorithm to find $\mathbf{e}$ given $\mathbf{s} = H \cdot (\mathbf{c} + \mathbf{e})$, whenever $\mathbf{e}$ does not have too many bits set.
- Given large H, usually very hard to find fast decoding algorithm.
- Use this difference in complexities for encryption.

Code-based encryption

- 1971 Goppa: Fast decoders for many matrices H.
- 1978 McEliece: Use Goppa codes for public-key cryptography.
  - Original parameters designed for 2^{64} security.
  - 2008 Bernstein–Lange–Peters: broken in $\approx 2^{60}$ cycles.
  - Easily scale up for higher security.
- 1986 Niederreiter: Simplified and smaller version of McEliece.
  - Public key: H with 1's on the diagonal.
  - Secret key: the fast Goppa decoder.
  - Encryption: Randomly generate $\mathbf{e}$ with t bits set. Send $H \cdot \mathbf{e}$.
  - Use hash of $\mathbf{e}$ to encrypt message with symmetric crypto (with 256 bits key).

Security analysis

- Some papers studying algorithms for attackers:
  1962 Prange; 1981 Omura; 1988 Lee–Brickell; 1988 Leon;
  1989 Krouk; 1989 Stern; 1989 Dumer; 1990 Coffey–Goodman;
  1990 van Tilburg; 1991 Dumer; 1991 Coffey–Goodman–Farrell;
  1993 Chabanne–Courteau; 1993 Chabaud; 1994 van Tilburg;
  1994 Canteaut–Chabanne; 1998 Canteaut–Chabaud;
  1998 Canteaut–Sendrier; 2008 Bernstein–Lange–Peters;
  2009 Bernstein–Lange–Peters–van Tilborg;
  2009 Bernstein (**post-quantum**); 2009 Finiasz–Sendrier;
  2010 Bernstein–Lange–Peters; 2011 May–Meurer–Thomae;
  2011 Becker–Coron–Joux; 2012 Becker–Joux–May–Meurer;
  2013 Bernstein–Jeffery–Lange–Meurer (**post-quantum**);
  2015 May–Ozerov.

Security analysis

- Some papers studying algorithms for attackers:
  1962 Prange; 1981 Omura; 1988 Lee–Brickell; 1988 Leon;
  1989 Krouk; 1989 Stern; 1989 Dumer; 1990 Coffey–Goodman;
  1990 van Tilburg; 1991 Dumer; 1991 Coffey–Goodman–Farrell;
  1993 Chabanne–Courteau; 1993 Chabaud; 1994 van Tilburg;
  1994 Canteaut–Chabanne; 1998 Canteaut–Chabaud;
  1998 Canteaut–Sendrier; 2008 Bernstein–Lange–Peters;
  2009 Bernstein–Lange–Peters–van Tilborg;
  2009 Bernstein (**post-quantum**); 2009 Finiasz–Sendrier;
  2010 Bernstein–Lange–Peters; 2011 May–Meurer–Thomae;
  2011 Becker–Coron–Joux; 2012 Becker–Joux–May–Meurer;
  2013 Bernstein–Jeffery–Lange–Meurer (**post-quantum**);
  2015 May–Ozerov.
- 256 KB public key for 2^{146} pre-quantum security.
- 512 KB public key for 2^{187} pre-quantum security.
- 1024 KB public key for 2^{263} pre-quantum security.

Security analysis

- Some papers studying algorithms for attackers:
  1962 Prange; 1981 Omura; 1988 Lee–Brickell; 1988 Leon;
  1989 Krouk; 1989 Stern; 1989 Dumer; 1990 Coffey–Goodman;
  1990 van Tilburg; 1991 Dumer; 1991 Coffey–Goodman–Farrell;
  1993 Chabanne–Courteau; 1993 Chabaud; 1994 van Tilburg;
  1994 Canteaut–Chabanne; 1998 Canteaut–Chabaud;
  1998 Canteaut–Sendrier; 2008 Bernstein–Lange–Peters;
  2009 Bernstein–Lange–Peters–van Tilborg;
  2009 Bernstein (**post-quantum**); 2009 Finiasz–Sendrier;
  2010 Bernstein–Lange–Peters; 2011 May–Meurer–Thomae;
  2011 Becker–Coron–Joux; 2012 Becker–Joux–May–Meurer;
  2013 Bernstein–Jeffery–Lange–Meurer (**post-quantum**);
  2015 May–Ozerov.
- 256 KB public key for 2^{146} pre-quantum security.
- 512 KB public key for 2^{187} pre-quantum security.
- 1024 KB public key for 2^{263} pre-quantum security.
- Post-quantum (Grover): below 2^{263}, above 2^{131}.

Binary Goppa code

Let $q = 2^m$. A binary Goppa code is often defined by

- a list $L = (a_1, \ldots, a_n)$ of n distinct elements in $\mathbb{F}_q$, called the support.
- a square-free polynomial $g(x) \in \mathbb{F}_q[x]$ of degree t such that $g(a) \neq 0$ for all $a \in L$. $g(x)$ is called the Goppa polynomial.
- E.g. choose $g(x)$ irreducible over $\mathbb{F}_q$.

The corresponding binary Goppa code $\Gamma(L, g)$ is

$$\left\{ \mathbf{c} \in \mathbb{F}_2^n \,\middle|\, S(\mathbf{c}) = \frac{c_1}{x - a_1} + \frac{c_2}{x - a_2} + \cdots + \frac{c_n}{x - a_n} \equiv 0 \bmod g(x) \right\}$$

- This code is linear $S(\mathbf{b} + \mathbf{c}) = S(\mathbf{b}) + S(\mathbf{c})$ and has length n.
- What can we say about the dimension and minimum distance?

Dimension of $\Gamma(L, g)$

- $g(a_i) \neq 0$ implies $\gcd(x - a_i, g(x)) = 1$, thus get polynomials

$$(x - a_i)^{-1} \equiv g_i(x) \equiv \sum_{j=0}^{t-1} g_{i,j} x^j \bmod g(x)$$

  via XGCD. All this over $\mathbb{F}_q = \mathbb{F}_{2^m}$.
- In this form, $S(\mathbf{c}) \equiv 0 \bmod g(x)$ means

$$\sum_{i=1}^{n} c_i \left(\sum_{j=0}^{t-1} g_{i,j} x^j \right) = \sum_{j=0}^{t-1} \left(\sum_{i=1}^{n} c_i g_{i,j} \right) x^j = 0,$$

  meaning that for each $0 \leq j \leq t-1$:

$$\sum_{i=1}^{n} c_i g_{i,j} = 0.$$

- These are t conditions over $\mathbb{F}_q$, so tm conditions over $\mathbb{F}_2$. Giving an $(n - tm) \times n$ parity check matrix over $\mathbb{F}_2$.
- Some rows might be linearly dependent, so $k \geq n - tm$.

Nice parity check matrix

Assume $g(x) = \sum_{i=0}^{t} g_i x^i$ monic, i.e., $g_t = 1$.

$$H = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ g_{t-1} & 1 & 0 & \dots & 0 \\ g_{t-2} & g_{t-1} & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ g_1 & g_2 & g_3 & \dots & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ a_1 & a_2 & a_3 & \cdots & a_n \\ a_1^2 & a_2^2 & a_3^2 & \cdots & a_n^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_1^{t-1} & a_2^{t-1} & a_3^{t-1} & \cdots & a_n^{t-1} \end{pmatrix}$$

$$\cdot \begin{pmatrix} \frac{1}{g(a_1)} & 0 & 0 & \dots & 0 \\ 0 & \frac{1}{g(a_2)} & 0 & \dots & 0 \\ 0 & 0 & \frac{1}{g(a_3)} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \frac{1}{g(a_n)} \end{pmatrix}$$

Minimum distance of $\Gamma(L, g)$. Put $s(x) = S(\mathbf{c})$

$$s(x) = \sum_{i=1}^{n} c_i/(x - a_i)$$

Minimum distance of $\Gamma(L, g)$. Put $s(x) = S(\mathbf{c})$

$$\begin{aligned} s(x) &= \sum_{i=1}^{n} c_i/(x - a_i) \\ &= \left(\sum_{i=1}^{n} c_i \prod_{j \neq i} (x - a_j) \right) / \prod_{i=1}^{n} (x - a_i) \equiv 0 \bmod g(x). \end{aligned}$$

- $g(a_i) \neq 0$ implies $\gcd(x - a_i, g(x)) = 1$, so $g(x)$ divides $\sum_{i=1}^{n} c_i \prod_{j \neq i} (x - a_j)$.
- Let $\mathbf{c} \neq 0$ have small weight $\mathrm{wt}(\mathbf{c}) = w \leq t = \det(g)$. For all i with $c_i = 0$, $x - a_i$ appears in every summand.

Minimum distance of $\Gamma(L, g)$. Put $s(x) = S(\mathbf{c})$

$$\begin{aligned} s(x) &= \sum_{i=1}^{n} c_i/(x - a_i) \\ &= \left(\sum_{i=1}^{n} c_i \prod_{j \neq i} (x - a_j) \right) / \prod_{i=1}^{n} (x - a_i) \equiv 0 \bmod g(x). \end{aligned}$$

- $g(a_i) \neq 0$ implies $\gcd(x - a_i, g(x)) = 1$, so $g(x)$ divides $\sum_{i=1}^{n} c_i \prod_{j \neq i} (x - a_j)$.
- Let $\mathbf{c} \neq 0$ have small weight $\mathrm{wt}(\mathbf{c}) = w \leq t = \det(g)$. For all i with $c_i = 0$, $x - a_i$ appears in every summand. Cancel out those $x - a_i$ with $c_i = 0$.
- The denominator is now $\prod_{i, c_i \neq 0} (x - a_i)$, of degree w.
- The numerator now has degree $w - 1$ and $\deg(g) > w - 1$ implies that the numerator is $= 0$ (without reduction mod g), which is a contradiction to $\mathbf{c} \neq 0$, so $\mathrm{wt}(\mathbf{c}) = w \geq t + 1$.

Better minimum distance for $\Gamma(L, g)$

- Let $\mathbf{c} \neq 0$ have small weight $\mathrm{wt}(\mathbf{c}) = w$.
- Put $f(x) = \prod_{i=1}^{n}(x - a_i)^{c_i}$ with $c_i \in \{0, 1\}$.
- Then the derivative $f'(x) = \sum_{i=1}^{n} c_i \prod_{j \neq i}(x - a_i)^{c_i}$.
- Thus $s(x) = f'(x)/f(x) \equiv 0 \bmod g(x)$.
- As before this implies $g(x)$ divides the numerator $f'(x)$.
- Note that over $\mathbb{F}_{2^m}$:

$$(f_{2i+1}x^{2i+1})' = f_{2i+1}x^{2i}, \quad (f_{2i}x^{2i})' = 0 \cdot f_{2i}x^{2i-1} = 0,$$

  thus $f'(x)$ contains only terms of even degree and $\deg(f') \leq w - 1$. Assume w odd, thus $\deg(f') = w - 1$.
- Note that over $\mathbb{F}_{2^m}$: $(x + 1)^2 = x^2 + 1$

Better minimum distance for $\Gamma(L, g)$

- Let $\mathbf{c} \neq 0$ have small weight $\mathrm{wt}(\mathbf{c}) = w$.
- Put $f(x) = \prod_{i=1}^{n}(x - a_i)^{c_i}$ with $c_i \in \{0, 1\}$.
- Then the derivative $f'(x) = \sum_{i=1}^{n} c_i \prod_{j \neq i}(x - a_i)^{c_i}$.
- Thus $s(x) = f'(x)/f(x) \equiv 0 \bmod g(x)$.
- As before this implies $g(x)$ divides the numerator $f'(x)$.
- Note that over $\mathbb{F}_{2^m}$:

$$(f_{2i+1}x^{2i+1})' = f_{2i+1}x^{2i}, \quad (f_{2i}x^{2i})' = 0 \cdot f_{2i}x^{2i-1} = 0,$$

  thus $f'(x)$ contains only terms of even degree and $\deg(f') \leq w - 1$. Assume w odd, thus $\deg(f') = w - 1$.
- Note that over $\mathbb{F}_{2^m}$: $(x + 1)^2 = x^2 + 1$ and in general

$$f'(x) = \sum_{i=0}^{(w-1)/2} F_{2i}x^{2i} = \left(\sum_{i=0}^{(w-1)/2} F_{2i}x^{i}\right)^2 = F^2(x).$$

- Since $g(x)$ is square-free, $g(x)$ divides $F(x)$, thus $w \geq 2t + 1$.

Decoding of in $\Gamma(L, g)$

- Decoding works with polynomial arithmetic.
- Fix $\mathbf{e}$. Let $\sigma(x) = \prod_{i,e_i \neq 0}(x - a_i)$. Same as $f(x)$ before.
- $\sigma(x)$ is called error locator polynomial. Given $\sigma(x)$ can factor it to retrieve error positions, $\sigma(a_i) = 0 \Leftrightarrow$ error in i.
- Split into odd and even terms: $\sigma(x) = a^2(x) + xb^2(x)$.
- Note as before $s(x) = \sigma'(x)/\sigma(x)$ and $\sigma'(x) = b^2(x)$.
- Thus

$$b^2(x) \equiv \sigma(x)s(x) \equiv (a^2(x) + xb^2(x))s(x) \bmod g(x)$$

$$b^2(x)(x + 1/s(x)) \equiv a^2(x) \bmod g(x)$$

- Put $v(x) \equiv \sqrt{x + 1/s(x)} \bmod g(x)$ (from syndrome $s(x)$), then $a(x) \equiv b(x)v(x) \bmod g(x)$.
- Use XGCD on v and g, stop part-way when

$$a(x) = b(x)v(x) + h(x)g(x),$$

with $\deg(a) \leq \lfloor t/2 \rfloor, \deg(b) \leq \lfloor (t-1)/2 \rfloor$.

More exciting codes

- Niederreiter's proposal was to use generalized Reed-Solomon codes, this was broken in 1992 by Sidelnikov and Shestakov.
- In general we distinguish between generic attacks (such as information-set decoding) and structural attacks (that use the structure of the code).
- Gröbner basis computation is a generally powerful tool for structural attacks.
- Cyclic codes need to store only top row of matrix, rest follows by shifts. Quasi-cyclic: multiple cyclic blocks.
- QC Goppa: too exciting, too much structure.
- Interesting candidate: Quasi-cyclic Moderate-Density Parity-Check (QC-MDPC) codes, due to Misoczki, Tillich, Sendrier, and Barreto (2012).
  Most recent proposal: QcBits by Tung Chou.
- Hermitian codes, general algebraic geometry codes.