Goppa Codes and the McEliece PKCS

Thomas Risse
Institute of Informatics & Automation, IIA
Faculty E&I, Hochschule Bremen, HSB

Subotica Tech, 3.-4.6.2010
Agenda

1. in retrospect: RSA, Diffie-Hellman, ElGamal
2. in prospect: quantum computers, QCs
3. bad news: QC-broken codes
4. good news: codes which are (not jet) QC-broken
5. (error correcting) Goppa Codes
6. McEliece public key cryptography
7. McEliece PKCS – some open problems
in retrospect: RSA

- RSA\textsuperscript{1} ’first’ (asymmetric) PKCS in 1978

A. publishes public key \((n, e)\) where \(n = pq\) with \(p\) and \(q\) prime, \(1 < e < \varphi(n)\), \(\gcd(e, \varphi(n)) = 1\) and Euler-function

\[
\varphi(n) := \left| \{1 \leq u \leq n, \gcd(u, n) = 1 \} \right| = (p - 1)(q - 1).
\]

A. keeps private key \(d\) with \(de = 1 \mod \varphi(n)\) as a secret.
B. encrypts message \(m\) to \(c = m^e \mod n\) and sends \(c\) to A.
A. receives \(c\) and decrypts to \(c^d \mod n = m^{de} \mod n = m\).

security = difficulty to factor large integers \(n = pq\) with primes \(p\) and \(q\), \(n\) is typically 1024–2048 bits long

factored challenges of RSA Laboratories:

Typically, RSA is used to exchange keys for some symmetric encryption/decryption method like AES.

in retrospect: Diffie-Hellman

- Diffie-Hellman(-Merkle)\(^2\) key exchange

  A. and B. agree to use prime \(p\) and base \(g\).
  A. chooses secret integer \(a\) and sends \(k_A = g^a \mod p\) to B.
  B. chooses secret integer \(b\) and sends \(k_B = g^b \mod p\) to A.
  A. computes common secret \(s = k_B^a \mod p = g^{ba} \mod p\).
  B. computes common secret \(s = k_A^b \mod p = g^{ab} \mod p\).

  security = difficulty to compute discrete logarithms
  Typically, \(p\) as well as \(a\) and \(b\) are large integers, with
  some 100 decimal digits.
  exchange e.g. keys of some symmetric
  encryption/decryption method like AES

\(^2\)W. Diffie, M.E. Hellman (1976): New Directions in Cryptography; IEEE
Transactions on Information Theory IT-22 (6): 644–654
in retrospect: ElGamal

- **El-Gamal\(^3\)** public-key cryptography based on Diffie-Hellman key exchange

  A. chooses cyclic group \((G, \ast)\) with \(|G| = q\) and \(G = \langle g \rangle\).
  A. chooses random integer \(0 \leq a < q\) as private key.
  A. computes \(h = g^a\) and publishes \(G, q, g, h\) as public key.
  B. chooses random integer \(0 \leq b < q\), computes \(c_1 = g^b\).
  B. computes common secret \(s = h^b = g^{ab}\).
  B. represents message \(m\) as element \(m' \in G\).
  B. computes \(c_2 = m' \ast s\) and sends \((c_1, c_2)\) to A.
  A. computes common secret \(s = c_1^a = g^{ba} = g^{ab} = h^b\).
  A. computes \(c_2 / s = m' \ast s / s = m'\) and converts \(m'\) to \(m\).

security = difficulty to compute discrete logarithms private keys per session!

---

\(^3\) T. ElGamal (1985): A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms; IEEE Transactions on Information Theory, v. IT-31, n. 4, pp469–472
in prospect: quantum computers, QCs

**Classical Computer**  The state is given by the $n$ bits of all registers.

**Quantum Computer**  The state is a quantum superposition of all possible states, i.e. $2^n$ for $n$ qubit registers. A quantum operation on the system simultaneously operates on all $2^n$ states, i.e. in one clock tick a quantum operation could compute $2^n$ machine states at once. There are quantum gates, quantum circuits, quantum computational models, quantum Turing machines . . .

in prospect: how do QCs store data?

see e.g. [3]

- A two-level spin-$\frac{1}{2}$ particle has to states, spin-down $|\downarrow\rangle$ and spin-up $|\uparrow\rangle$, or $|0\rangle$ and $|1\rangle$, respectively, described by the wave function $\psi = \alpha|0\rangle + \beta|1\rangle$ where $\alpha, \beta \in \mathbb{C}$, i.e., a linear superposition amongst any of the possible ’classical’ states of the system.
- $|\alpha|^2$ and $|\beta|^2$ with $|\alpha|^2 + |\beta|^2 = 1$ are the probabilities for finding the particle in the corresponding state.
- A set of $k$ spin-$\frac{1}{2}$ particles is in $2^k$ basis states corresponding to the $2^k$ possible bit-strings of length $k$. 
in prospect: how do QCs operate?

- Any operation on a single qubit is modelled by a unitary transformation $U$, e.g. $U_\theta = \begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix}$ is applied to the coefficients $\alpha$ and $\beta$, e.g. $U_\pi |0\rangle = -|1\rangle$ and $U_\pi |1\rangle = |0\rangle$ flips a qubit.

- Other unitary transformations operate on several qubits. Writing the two-particle basis states as
  
  $|00\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$, $|01\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$, $|10\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$, $|11\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$

  the unitary transformation $U_{\text{XOR}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$

  implements the XOR-operation if $|A\rangle = 1$ $|B\rangle = \text{NOT} |B\rangle$. 
in prospect: QCs performance

According to [5] to factor a big integer

- a classical computer takes

<table>
<thead>
<tr>
<th>year</th>
<th># bits</th>
<th>1024</th>
<th>2048</th>
<th>4096</th>
</tr>
</thead>
<tbody>
<tr>
<td>2006</td>
<td>$10^5 a$</td>
<td>$5 \times 10^{15} a$</td>
<td>$3 \times 10^{29} a$</td>
<td></td>
</tr>
<tr>
<td>2024</td>
<td>$38a$</td>
<td>$10^{12} a$</td>
<td>$7 \times 10^{25} a$</td>
<td></td>
</tr>
<tr>
<td>2042</td>
<td>$3d$</td>
<td>$3 \times 10^{8} a$</td>
<td>$2 \times 10^{22} a$</td>
<td></td>
</tr>
</tbody>
</table>

- a quantum computer takes

<table>
<thead>
<tr>
<th>year</th>
<th># bits</th>
<th>1024</th>
<th>2048</th>
<th>4096</th>
</tr>
</thead>
<tbody>
<tr>
<td>near (?) future</td>
<td>4.5’</td>
<td>36’</td>
<td>4.8h</td>
<td></td>
</tr>
</tbody>
</table>

by Shor’s algorithm [10], [3]

involving the quantum Fourier transform
bad news: QC-broken codes

according to [1]

- RSA public key encryption
- Diffie-Hellman key-exchange
- Elliptic curve cryptography
- Buchmann-Williams key exchange
  (based on the difficulty to solve Pell’s equation
  $x^2 - d y^2 = 1$ for fixed $d \in \mathbb{N}$, $\sqrt{d} \notin \mathbb{N}$
  which is at least as hard as factoring integers!)
- algebraically homomorphic
good news: codes which are (not jet) QC-broken

according to [1]

- McEliece PKCS
- GGH (Goldreich, Goldwasser, Halevi)
  = lattice analogon to McEliece
  see NTRU Cryptosystems Inc.: N-th degree truncated polynomial ring, i.e. \( R[x]/(x^N - 1) \) where \( R = \mathbb{Z}/p\mathbb{Z} \),
  now www.securityinnovation.com
- lattice based cryptosystems
- hash based digital signature schemes
- multivariate public key cryptography
(error correcting) Goppa codes

Def. Given a Goppa polynomial \( g(x) \in \mathbb{GF}(q^m)[x] \), \( \deg(g) = t \) and \( L = \{\lambda_1, \ldots, \lambda_n\} \subset \mathbb{GF}(q^m) \) with \( g(\lambda_i) \neq 0 \) for \( i = 1, 2, \ldots, n \). Then
\[
\Gamma(L, g) = \{(c_1, \ldots, c_n) \in \mathbb{GF}_q^n : \sum_{i=1}^n \frac{c_i}{x - \lambda_i} = 0 \bmod g(x)\}
\]
is a (classic) \([n, k, d]\) Goppa code.

- alternate generalized Reed-Solomon code \( k \geq n - mt \)
- \( d \geq t + 1 \)
  if the code in addition is binary and \( g \) without multiple zeroes then \( d \geq 2t + 1 \);
- for suitable Goppa polynomial \( g \), e.g. \( g(x) = x^{d-1} \) the code \( \Gamma(L, g) \) is cyclic;
- there is a sequence of Goppa codes which asymptotically meets the Gilbert-Varshamov bound.
- decoding by extended Euclid, by Berlekamp-Massey, by threshold or by list decoding ...
McEliece PKCS – keys

private keys

Choose three binary matrices

- $S$, a random non-singular $k \times k$ scrambler matrix
- $P$, a random $n \times n$ permutation matrix
- $G$, the $k \times n$ generator matrix of a $[n, k, 2t + 1]$-Goppa-Code, i.e. $n = 2^m$, $k = n - mt$

public keys

To the private key $(S, G, P)$ the public key $\hat{G} = SGP$ with $t' \leq t$ belongs.
McEliece PKCS – method

Encryption
The plain text is partitioned into bit-blocks $u$ of length $k$. Each block is Goppa coded to $c = u \hat{G}$. The code word $c$ of length $n$ is charged with $t' \leq t$ 1-bit-errors, i.e. modified to $y = c + e$ with error vector $e$.

Decryption
Using the private key, the receiver decodes $yP^{-1} = cP^{-1} + eP^{-1} = uSG + eP^{-1}$ per fast Goppa decoding to $uS$ and per $uSS^{-1}$ to $u$.

Security
Decoding of linear codes is NP-complete [9].

Typical parameters
[8] $n = 1024, k = 524, t = 50$; [2] $n = 2960, k = 2288, t = 56$

Drawback
very long keys, e.g. [2] 1537536 bits $\approx$ 187KB
Goppa Codes & McEliece PKCS – open problems

- which Goppa codes/polynomials over which fields are suitable?
- which parameters guarantee which degree of security?
- implementations of which codes are fast and at the same time secure?
- which implementations optimize space, speed and/or energy consumption?
References I


References II


