# McEliece Cryptosystem

## An overview

Jannis Priesnitz

University of Applied Sciences Darmstadt
Department of Computer Science
Schöfferstraße 3
64295 Darmstadt

March 14, 2017

# Outline

1. Post quantum cryptography
   - Post quantum computing
   - Shors algorithm
   - Canditates for post quantum cryptography
2. McEliece Cryptosystem
   - Basics
   - Key generation
   - Encryption
   - Decryption
   - Signature creation
   - Key space
   - Correctness
   - Security properties
3. Variants of McEliece
   - Niederreiter cryptosystem
   - Wild McEliece

# Quantum computing

What is quantum computing?

- Computer based on principle of quantum mechanics
- Quantum computer contains qbits instead of bit
  - Every qbit can be in state 0 or 1 but can also be in every *sperposition* in between
  - The state is destroyed by reading it (see Schrödingers cat)
- They are able to solve computational strong problems efficient

# Quantum computing

How real is it?

- 2001: IBM Almaden Research Center realized a system with 7 Qubits
  - Factored 15 into it's prime factors 3 und 5
- ...
- 2013: D-Wave Systems sells first quantum computes to Google and NASA

*We can say that quantum computing is a huge game changer on the filed of computation.*
*We can't say how real they are.*

# Cryptography pre- and post quantum

What is changing with focus on cryptography?

- Asymmetric state of the art security is no longer secure
  - RSA $\rightarrow$ prime factorization
  - ECC $\rightarrow$ discrete logarithm problem
- Symmetric algorithms are still secure

*We need other asymmetric cryptographic schemes than the established ones.*

# Post quantum cryptography

- Created by Daniel J. Bernstein
- Algorithm loosing *almost* no security executed on a quantum computer
- Completely different mathematical base than established ones

# Shors algorithm

- Developed 1994 by Peter Shor
- Solves prime factorization and discrete logarithm problem efficiently
- Monte Carlo Algorithm
- Classical part
  - Mainly calculating gcd
- Quantum part
  - Mainly quantum Fouriertransformation

# Classical part

For n as composed number:

```
start:
 select an integer $1 < x < n$
 if gdc(x,n) is 1 // Euclidian algorithm
  return 1
 else
  r = compute_order(x) // quantum part
  if r is odd or x^(r/2) is equivalent -1(mod n)
   goto start
  else
   return gcd(x^(r/2) - 1, n)
```

# Quantum part (sketch)

For n as input from the classical part:

```
start:
Determine q as power of 2 with n^2 <= q <= 2n^2
Init the input register with superposition of a mod q
Init the output  register with x^a(mod n)
Perform quantum Fouriertransformation on input register
r = meassurement of output register
if r != order(x)
 goto start
else
 return r
```

Note: The input quantum reg has all possible states of a mod.

# Complexity considerations

- $O((logn)^3)$ Instructions
- Complexity class of BQP
-

# Lattice-based cryptography

- Came um in 1996
- Lattice over a n-dimensional finite Euclidian field
    - Strong peridicity required
- Set of vectors setting up the base
- Unique representation
- Cryptographic problem: Finding closest vector to an lattice point
-

# Multivariate cryptography

- First mentione 1988
- Multivariate polynomials over a finite field F
- Defined over both a ground and an extension field
- Promising for digital signatures
- Private key consist of two affine transformations having an group endomorphism
- Public key is the concatination of them

# Hash-based cryptography

- Created by Lamport and Merkle in 1979
- Only usable for digital signatures
- Hash-based cryptographic algorithms
  - PQ resistance required
- Limited count of signatures for one key

# Code-based cryptography

- Founded by Robert McEliece in 1978
- Based on error correcting codes
  - Goppa Codes
  - Irreducibility
- Good for encryption
- Difficult for signing

# McEliece Cryptosystem

- Binary, irreducibility Goppa codes $C$
  - Length: $n = 2^m$
  - Dimension of $k >= n - tm$
  - Correct up to $t$ errors
- Irreducible polynomial
  - Degree: $t$ over $GF(2^m)$.
  - Dimension of $k >= n - tm$

# Establishing a key pair

- Select $n$ and $t$ which defines the code
- Select an polynomial of degree $t$ over $GF(2^m)$
- Test $t$ if it is irreducible
  - repeat if it is reducible
- Produce a $n \times k$ generator matrix $G$
- To improve the efficiency $G$ can be transformed into canonical form

# Establishing a key pair 2

- Camouflage $G$
  - Select $S$
    - Random dense $k \times k$ matrix
    - Nonsingular "scrambling"
  - Select Permutationmatrix $P$
    - Random $n \times n$ matrix
  - Compute $G' = SGP$
    - Same rate and distance like $G$

# Encrypting a message

- Divide message into k-bit blocks $u$
- $x = uG' + z$
    - $z$ is a random vector with length $n$ and weight $t$
- $x$ will be transmitted encrypted to the key owner

# Decrypting a message

With $x$ as received cipher text message block:

- Eliminate P: $x' = xP^{-1}$
  - $P^{-1}$ inverse of permutation matrix
- Perform correcting algorithm for $C$
  - Codeword $u'$ next to $x'$ is calculated
  - Suggested error correction: Algorithm of Patterson
- Get the plaintext $u = u'S^{-1}$
  - Eliminate

# Issues with creating a signature

-

# Solutions

- test

# Lenght of cryptographically strong keys

- test

# Correctness of the presented scheme

- test

# Security properties

- test

# Introduction to Niederreither cryptosystem

- test

## test

- test