Technische Universiteit Eindhoven University of Technology

Attacking the McEliece cryptosystem

Christiane Peters

Public-key cryptography

- Use a key pair consisting of a public key and a private key. Each key has only one task: the public key is used for encryption and the private key for decryption.
- Should be infeasible to derive the private key from the public key.
- In practice: look up the public key, i.e., on a key server, a website etc. and encrypt message;
  send the encrypted message to the owner of the private key who is able to decrypt the message using the private key.

Post-quantum cryptography

- Quantum computers will break the most popular public-key cryptosystems (PKCs).
- Post-quantum cryptography—a very recent field of cryptography—deals with cryptosystems that run on conventional computers and are secure against attacks by quantum computers.
- The McEliece cryptosystem—introduced by R.J McEliece in 1978—is one of the public-key systems without known vulnerabilities to attacks by quantum computers.

Idea behind the McEliece PKC

- Based on algebraic coding theory
- The public key in McEliece's system is a random-looking algebraic code over a finite field
- Encryption in McEliece's system is remarkably fast:
  the sender simply encodes a plaintext and adds some errors
- The receiver, having generated the code by secretly transforming a Goppa code, can use standard Goppa-code decoders to correct the errors and recover the plaintext.

Set-up of the McEliece PKC

- Given a 50-error correcting classical binary Goppa code Γ of length 1024 and dimension 524 which is kept secret.
- The McEliece secret key is a triple (G, S, P) consisting of a generator matrix G for the code Γ; a 1024×1024 permutation matrix P, and an invertible 524×524 matrix S;
- The sizes $1024, 524, 50$ are public system parameters.
- The McEliece public key is the 524×1024 matrix $\hat{G} = SGP$.

Encryption and Decryption

McEliece encryption of a message $\mathbf{m} \in \{0,1\}^{524}$:

- Compute $\mathbf{m}\hat{G}$ and hide the message by adding a random length-1024 error vector $\mathbf{e}$ of weight 50.
- Send $\mathbf{y} = \mathbf{m}\hat{G} + \mathbf{e}$.

McEliece decryption:

- Compute $\mathbf{y}P^{-1} = \mathbf{m}SG + \mathbf{e}P^{-1}$.
- Note that $\mathbf{m}SG$ is a codeword in Γ and that the permuted error vector $\mathbf{e}P^{-1}$ has weight 50.
- Use the decoding algorithm to find $\mathbf{m}S$ and thereby $\mathbf{m}$.

The McEliece PKC from an attacker's point of view

Two possible attacks:

- Find out secret code; i.e., find G given $\hat{G}$.
- Or decode $\mathbf{y}$ without knowing an efficient decoding algorithm for the public code given by $\hat{G}$.

Attacks on the McEliece PKC

Most effective attack against the McEliece PKC is information-set decoding; used for decoding a given number of errors in $\mathbf{y}$ without knowledge of a decoding algorithm.

- Many variants: McEliece (1978), Leon (1988), Lee and Brickell (1988), Stern (1989), van Tilburg (1990), Canteaut et al. (1994 and 1998).
- Our attack is most easily understood as a variant of Stern's attack.
- Our attack is faster by a factor of more than 150 than previous attacks; now within reach of a moderate cluster of computers.

Running time in practice

- Attack on a single computer with a 2.4GHz Intel Core 2 Quad Q6600 CPU would need approximately 1400 days (2^{58} CPU cycles).
- Running the software on 200 such computers would reduce the average time to one week.
- Canteaut et al (1998): implementation on a 433MHz DEC Alpha CPU; one such computer would need approximately 7400000 days (2^{68} CPU cycles).
- Note: hardware improvements (DEC Alpha to Core 2) only reduce 7400000 days to 220000 days. Remaining speedup factor of 150 comes from our improvements of the attack itself.

First successful attack

We were able to extract a plaintext from a ciphertext by decoding 50 errors in a $[1024, 524]$ binary code.

- There were about 200 computers involved, with about 300 cores
- Computation finished in under 90 days
  (most of the cores put in far fewer than 90 days of work; some of which were considerably slower than a Core 2)
- Used about 8000 core-days
- Error vector found by Walton cluster at SFI/HEA Irish Centre of High-End Computing (ICHEC)
- Improved attack such that only 5000 core-days would be needed on average

Conclusions

- Our attack demonstrated that the parameters were chosen too small.
- It should not be interpreted as destroying the McEliece cryptosystem.
- In fact, the best known attacks are exponential in the main parameter and thus larger parameters lead to secure systems.

Literature

- Daniel J. Bernstein, Tanja Lange, Christiane Peters. *Attacking and defending the McEliece cryptosystem.* In: Post-Quantum Cryptography, LNCS Vol. 5299, pp. 31–46. Springer, 2008.
- Robert J. McEliece. *A Public-Key Cryptosystem Based on Algebraic Coding Theory.* Jet Propulsion Laboratory DSN Progress Report 42–44. 1978.
- Raphael Overbeck and Nicolas Sendrier. *Code-based cryptography.* In: Post-quantum cryptography, In Daniel J. Bernstein, Johannes Buchmann, and Erik Dahmen, eds., Introduction to Post-Quantum Cryptography. Springer, 2009.

This is joint work with Daniel J. Bernstein (University of Illinois at Chicago) and Tanja Lange (TU/e).