McEliece Cryptosystem in real life: security and implementation

Bhaskar Biswas and Nicolas Sendrier

SECRET - INRIA Rocq.

Some numbers!

scheme	key gen	enc cycle	dec cycle
Ntru	203983313	894427	1617090
RSA1-1024	188582730	225593	6240622
RSA1-2048	1147314285	431452	30819975
RSA2-1024	190663380	195172	6155468
RSA2-2048	1179499485	403327	30703080

Table: Selected parameters at a glance

Some numbers! cont...

m	t	key size (Mbits)	block length	cycles per byte encr.	cycles per byte decr.	security
11	32	.647	2048	516	5548	88.82
11	44	.860	2048	623	7921	99.87
12	21	.967	4096	433	3885	88.74
12	26	1.19	4096	474	4991	100.97
12	40	1.79	4096	577	7040	128.79
13	17	1.71	8192	414	5522	89.82
13	20	2.01	8192	445	6560	100.30
13	29	2.90	8192	506	9649	129.03

Table: Selected parameters at a glance

Motivation

- Old and considered fast but implementation of McEliece public-key system never been thoroughly studied.

Motivation

- Old and considered fast but implementation of McEliece public-key system never been thoroughly studied.
- Consider the problem and provide a careful implementation together with cryptanalysis.

Motivation

- Old and considered fast but implementation of McEliece public-key system never been thoroughly studied.
- Consider the problem and provide a careful implementation together with cryptanalysis.
- Provide a reference for measuring speed and scalability.

Motivation

- Old and considered fast but implementation of McEliece public-key system never been thoroughly studied.
- Consider the problem and provide a careful implementation together with cryptanalysis.
- Provide a reference for measuring speed and scalability.
- Compare with other, number-theory based, public key schemes.

Chronology

- Proposed in 1978 by Robert J. McEliece.

Chronology

- Proposed in 1978 by Robert J. McEliece.
- Niederreiter proposed a variance! Knapsack type. 1986.

Chronology

- Proposed in 1978 by Robert J. McEliece.
- Niederreiter proposed a variance! Knapsack type. 1986.
- Decoding Attacks:

Chronology

- Proposed in 1978 by Robert J. McEliece.
- Niederreiter proposed a variance! Knapsack type. 1986.
- Decoding Attacks:
  Lee and Brickell - Eurocrypt-'88.
  Leon - ITIT-'88.
  H. van Tilborg - CRYPTO-'88.
  J. Stern - Coding theory and applications-'89.
  A. Canteaut and F. Chabaud - '95.

Chronology

- Proposed in 1978 by Robert J. McEliece.
- Niederreiter proposed a variance! Knapsack type. 1986.
- Decoding Attacks:
  Lee and Brickell - Eurocrypt-'88.
  Leon - ITIT-'88.
  H. van Tilborg - CRYPTO-'88.
  J. Stern - Coding theory and applications-'89.
  A. Canteaut and F. Chabaud - '95.
- N. Courtois and M. Finiasz and N. Sendrier - Signature Scheme. 2001.

Problem statement

- Implementing a (public key) cryptosystem is a tradeoff between security and efficiency.

Problem statement

- Implementing a (public key) cryptosystem is a tradeoff between security and efficiency.
- Large public key, but the McEliece cryptosystem has a good security reduction and low complexity algorithms for encryption and decryption.

Problem statement

- Implementing a (public key) cryptosystem is a tradeoff between security and efficiency.
- Large public key, but the McEliece cryptosystem has a good security reduction and low complexity algorithms for encryption and decryption.
- We present a slightly modified version of the scheme (which we call *hybrid*).

Problem statement

- Implementing a (public key) cryptosystem is a tradeoff between security and efficiency.
- Large public key, but the McEliece cryptosystem has a good security reduction and low complexity algorithms for encryption and decryption.
- We present a slightly modified version of the scheme (which we call *hybrid*).
- Two modifications, increases the information rate by putting some data in the error pattern and reduces the public key size by making use of a generator matrix in row echelon form.

System description

The McEliece encryption scheme, as any public key encryption scheme, has to be described by a triple of procedures:

- a key generation procedure,

System description

The McEliece encryption scheme, as any public key encryption scheme, has to be described by a triple of procedures:

- a key generation procedure,
- an encryption procedure,

System description

The McEliece encryption scheme, as any public key encryption scheme, has to be described by a triple of procedures:

- a key generation procedure,
- an encryption procedure,
- a decryption procedure.

System description

We define an injective mapping $\varphi : \{0,1\}^\ell \to W_{n,t}$. Both φ and φ^{-1} should be easy to compute.

- System parameters: two integers m and t. Let $n = 2^m$ and $k = n - tm$.

System description

We define an injective mapping $\varphi : \{0,1\}^\ell \to W_{n,t}$. Both φ and φ^{-1} should be easy to compute.

- System parameters: two integers m and t. Let $n = 2^m$ and $k = n - tm$.
- Key generation:
  - generate a support $L = (\alpha_1, \ldots, \alpha_n)$ of n distinct elements of $\mathbf{F}_{2^m}$,
  - generate a monic irreducible generator polynomial $g(z) \in \mathbf{F}_{2^m}[z]$ of degree t.
  - The secret key is the pair (L, g) (i.e. the Goppa code $\Gamma(L, g)$ and its decoder)
  - The public key is a binary $k \times (n-k)$ matrix R where $G = (Id \mid R)$ is a generator matrix of $\Gamma(L, g)$ in row echelon form.

System description...

- Encryption: the plaintext is in $\{0,1\}^k \times \{0,1\}^\ell$ and the ciphertext in $\{0,1\}^n$

$$\begin{array}{ccc} \{0,1\}^k \times \{0,1\}^\ell & \longrightarrow & \{0,1\}^n \\ (x, x') & \longmapsto & E_G(x, \varphi(x')) \end{array}$$

System description...

- Encryption: the plaintext is in $\{0,1\}^k \times \{0,1\}^\ell$ and the ciphertext in $\{0,1\}^n$

$$\begin{array}{ccc} \{0,1\}^k \times \{0,1\}^\ell & \longrightarrow & \{0,1\}^n \\ (x, x') & \longmapsto & E_G(x, \varphi(x')) \end{array}$$

- Decryption: the ciphertext has the form $y = xG + e$, with $e = \varphi(x')$ of Hamming weight $\leq t$. Applying the decoder of $\Gamma(L, g)$ on y will provide x and $x' = \varphi^{-1}(e)$.

System description...

- Encryption: the plaintext is in $\{0,1\}^k \times \{0,1\}^\ell$ and the ciphertext in $\{0,1\}^n$

$$\begin{array}{ccc} \{0,1\}^k \times \{0,1\}^\ell & \longrightarrow & \{0,1\}^n \\ (x, x') & \longmapsto & E_G(x, \varphi(x')) \end{array}$$

- Decryption: the ciphertext has the form $y = xG + e$, with $e = \varphi(x')$ of Hamming weight $\leq t$. Applying the decoder of $\Gamma(L, g)$ on y will provide x and $x' = \varphi^{-1}(e)$.

There are two differences compared with the original system:

- We use the error to encode information bits.

System description...

- Encryption: the plaintext is in $\{0,1\}^k \times \{0,1\}^\ell$ and the ciphertext in $\{0,1\}^n$

$$\begin{array}{ccc} \{0,1\}^k \times \{0,1\}^\ell & \longrightarrow & \{0,1\}^n \\ (x, x') & \longmapsto & E_G(x, \varphi(x')) \end{array}$$

- Decryption: the ciphertext has the form $y = xG + e$, with $e = \varphi(x')$ of Hamming weight $\leq t$. Applying the decoder of $\Gamma(L, g)$ on y will provide x and $x' = \varphi^{-1}(e)$.

There are two differences compared with the original system:

- We use the error to encode information bits.
- We use a public key in row echelon form.

System description...

- Encryption: the plaintext is in $\{0,1\}^k \times \{0,1\}^\ell$ and the ciphertext in $\{0,1\}^n$

$$\begin{array}{ccc} \{0,1\}^k \times \{0,1\}^\ell & \longrightarrow & \{0,1\}^n \\ (x, x') & \longmapsto & E_G(x, \varphi(x')) \end{array}$$

- Decryption: the ciphertext has the form $y = xG + e$, with $e = \varphi(x')$ of Hamming weight $\leq t$. Applying the decoder of $\Gamma(L, g)$ on y will provide x and $x' = \varphi^{-1}(e)$.

There are two differences compared with the original system:

- We use the error to encode information bits.
- We use a public key in row echelon form.

Those changes improve credentiality and as we will see, have no real impact on security.

One way encryption scheme

Definition (OWE)

A public key encryption scheme is a *One Way Encryption* scheme if the probability of success of any of its adversary running in polynomial time is negligible.

One way encryption scheme

Definition (OWE)

A public key encryption scheme is a *One Way Encryption* scheme if the probability of success of any of its adversary running in polynomial time is negligible.

In practice, one needs more than just an OWE scheme.

- McE, though it is OWE, is vulnerable to many attacks.
- Given perfect hash functions exists, there are generic conversions, which, starting from an OWE scheme, provide a scheme resistant against adaptative chosen ciphertext attack.

One way encryption scheme

Definition (OWE)

A public key encryption scheme is a *One Way Encryption* scheme if the probability of success of any of its adversary running in polynomial time is negligible.

In practice, one needs more than just an OWE scheme.

- McE, though it is OWE, is vulnerable to many attacks.
- Given perfect hash functions exists, there are generic conversions, which, starting from an OWE scheme, provide a scheme resistant against adaptative chosen ciphertext attack.

We can prove, given the two following assumptions, the hybrid McEliece encryption scheme is OWE.

Security assumptions

- Hardness of decoding in the average case: The success probability

$$\Pr_{\Omega}\left(\mathcal{A}(xG+e,G)=(x,e)\right)\left|\Omega=\{0,1\}^k\times W_{n,t}\times\mathcal{M}_{k\times n}\right.$$

  of any adversary $\mathcal{A}$ running in polynomial time is negligible.
- Pseudo-randomness of binary Goppa Codes: there exists no efficient distinguisher for Goppa codes. In other words, the generator matrix of a Goppa code looks random.

The hybrid scheme is one way

Theorem

The hybrid McEliece scheme is one-way.

Proof omitted.

Key generation and Encryption

Recall, we generate the support L and generator polynomial g.

- The pair $\{L, g\}$ is the private key(s).
- The public keys is R, a binary $k \times (n-k)$ matrix, where $G = (Id \mid R)$ is a generator matrix of $\Gamma(L, g)$ in row echelon form.

Recall again, encryption is computed as,

$$(x, x') \longmapsto E_G(x, \varphi(x'))$$

Constant weight encoding

Prof. Nicolas Sendrier is going to deliver a talk on this part.
Il vaut mieux prier Dieu que ses saints!!!.

Decryption

Decryption consist of 2 distinct stages.

- Decoding of Goppa code to generate the error positions.
  Next 2 slides!
- Retrieval of plain text.
  This is the inverse function applied to retrieve the message.
  Where, $x' = \varphi^{-1}(e)$.

Goppa code decoding

The 3 stages involved in the decoding process, are,

- The syndrome computation.
- Solving the key equatoin.
- Computation of roots.

We precompute all the $f_j(z) = (z - \alpha_j)^{-1} \mod g(z)$. The syndrome of the word $a = (a_1, \ldots, a_n)$ is the sum

$$R_a(z) = \sum_{j=1}^{n} f_j(z) = \sum_{j=1}^{n} \frac{a_j}{z - \alpha_j} \mod g(z)$$

Patterson algorithm and Computation of roots

- The key equation is solved by Patterson algorithm. We chose to precompute the square roots modulo $g(z)$ (that is the $s_i(z)$ such that $s_i(z)^2 = z^i \mod g(z)$, $0 \leq i < t$).
- The roots of the locator polynomial are found by an exhaustive search on the field elements.

Simulation results

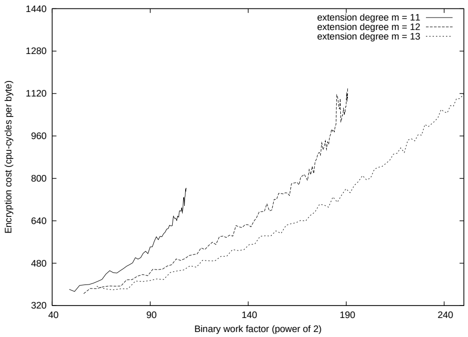

Figure: Encryption cost vs binary work factor for different extension degrees

Simulation results

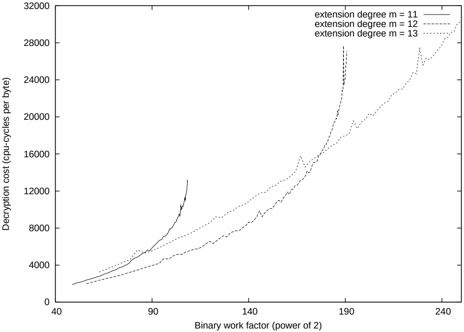

FIGURE: Decryption cost vs binary work factor for different extension degrees

Conclusion

We presented here a new modified version of McEliece cryptosystem and its full implementation. We have shown that code-based public key encryption scheme compares favorably with optimized implementation of number theory based schemes.
We plan to explore possibilities to improve the implementation, as well as widening the scope of parameters in our simulations.

Thank you! Questions?

Questions and remarks....shoot please!