The Goppa-Based McEliece Cryptosystem: Design and Security

Kalina Dimovska

Cryptography plays a crucial role in our everyday lives, ensuring the security of our digital communications and information. However, the rise of quantum computing poses a potential threat to current encryption methods like the RSA or ECC, which can be broken using Shor's algorithm. Hence, it is essential to explore some alternatives. In light of this, the thesis explores code-based cryptography, focusing primarily on the McEliece cryptosystem as a candidate for post-quantum cryptography.

Firstly, we delve into the fundamentals of coding theory and highlight the significance of Goppa codes in the McEliece system. We say $\mathcal{C}$ is a code over an arbitrary set A if $\mathcal{C} \subseteq A^*$. Controling errors in data being transmitted over a noisy channel is achieved by using error-correcting codes. During transmission, some of the coordinates of a message $\mathbf{m} \in A^k$ might get corrupted. To deal with this, we add some redundancies to the message $\mathbf{m} \mapsto \mathbf{c} \in A^n$ for $n > k$. The redundancies help the receiver of $\mathbf{y} = \mathbf{c} + \mathbf{e}$ to detect if any errors $\mathbf{e}$ have occurred in order to recover the original message. If a code $\mathcal{C}$ is a k-dimensional linear subspace of $\mathbb{F}_q^n$, then we call it an $[n, k]$ linear code. For a linear code we can define a matrix G whose rows give a basis of the code and call it a generator matrix. We can also define a parity-check matrix H of which $\mathcal{C}$ is the kernel. To encode a message $\mathbf{m}$ using linear codes, we just calculate $\mathbf{c} = \mathbf{m}G$.

In its original construction, the McElliece system uses irreducible binary Goppa codes, a type of linear codes based on algebraic curves. We denote a Goppa code by $\Gamma(g, L)$ where $g(x) \in \mathbb{F}_{2^m}[x]$ is a degree t polynomial for some fixed m and $L = \{\alpha_1, ..., \alpha_n\} \subseteq \mathbb{F}_{2^m}$ is a set of points which are not roots of the polynomial. Knowing the generator polynomial g for such a code is necessary and sufficient for efficient error correction and this fact will be used to construct the McElliece system. Practical examples are provided to illustrate the importance of both coding theory and algebra in understanding this cryptosystem. In particular, we calculate the parity-check matrix of a Goppa code with parameters $L = \mathbb{F}_{2^3}$ and $g(x) = x^2 + x + \alpha^3$ and then, with the help of MATLAB, we get the generator matrix of Γ. We also take a look at Patterson's algorithm for decoding Goppa codes and how to apply it in practice.

To construct the McEliece cryptosystem, we start with fixed parameters $n, t \in \mathbb{N}$, $t << n$, then choose an $[n, k]$ binary Goppa code and random matrices S and P having some additional properties. The public key is (G_{pub}, t) where $G_{pub} := SGP$ and the private key is $(S, \mathcal{D}, P)$, where $\mathcal{D}$ is an efficient decoding algorithm for Γ. After applying matrices S and P in the Mceliece key generation, the structure of the Goppa code is seemingly lost, that is, an attacker has to decode random-looking code which is NP-hard. The encryption is then done by transforming the message $\mathbf{m}$ into ciphertext $\mathbf{y} = \mathbf{m}G_{pub} + \mathbf{e}$, which is then decrypted by calculating $\mathbf{y}P^{-1} = \mathbf{m}G_{pub}P^{-1} + \mathbf{e}P^{-1} = \mathbf{m}SG + \mathbf{e}P^{-1}$ which can then, with the help of Patterson's algorithm, be decoded to get back $\mathbf{m}$. Again, we give a practical example in MATLAB of McEliece key generation, encryption and decryption.

Security of the cryptosystem is a priority when selecting parameters and implementing the McEliece system. The originally suggested parameters $(n, k, t) = (1024, 524, 50)$ are no longer secure, and recent research suggests new and improved parameter choices, for $(2960, 2288, 56)$ which achieve 128-bit

security, pointing out that the optimal rate is $R \approx 0.8$. We conclude that it is crucial to select the right parameters, keep key components secret, and use an IND-CCA2 secure variant of the McEliece system to avoid private-key or message-resend attacks and ensure security. While various ciphertext only attacks on the McEliece system have been studied, the Information-Set Decoding (ISD) attack emerges as the most effective. Still, all known ciphertext only attacks requires exponential time to be executed.

When it comes to practical use, the primary limitation of the McEliece system lies in its large key size. For example, to achieve the same security level, an instance of the McEliece can have a ≈ 192 KB size key, while an ECC one a 242 bit key, making the McEliece key more than 6353 times larger. Some efforts have been made to deal with this issue by replacing Goppa codes in the McElliece system with other code families, but most of these attempts have been shown to be insecure. The most promising solution currently seems to be using Goppa codes over larger fields. Going from binary Goppa codes to codes of the form $\Gamma_{31}(L, g)$ significantly reduces the key size while preserving the same security level. An additional suggestion is to use so-called wild Goppa codes $\Gamma_q(L, g^{q-1})$.

Upon comparing the McEliece system with other encryption methods (IKKR, Niederreiter, RSA, ECC), we can confirm that its key size is a notable challenge. However, it has a big advantage in terms of encryption and decryption speed, making it a good choice for certain practical applications where memory constraints are not a concern, but speed is important.

As we approach the era of quantum computing, the McEliece system deserves further exploration and consideration for its potential as a post-quantum encryption solution.