Summary of A Public-Key Cryptosystem Based on Algebraic Coding Theory

Kalle Nordgren kalle.nordgren.8070@student.uu.se
Hamit Efe Çınar hamit-efe.cinar.3004@student.uu.se
Tove Jansson tove.jansson.7568@student.uu.se
Arveen Emdad arveen.emdad.8201@student.uu.se

February 2024

1 Introduction

The author starts the paper by proposing an approach of the public-key cryptosystem, which is said to be ideal for distribution of obtained data from space by NASA over multi-user communication networks. [1] The approach is mentioned to replace a periodic distribution of a secret key to the sender and receiver and instead using theory based on algebraic codes for a public key.

2 Background

The author mentions multiple previous works of cryptosystems which can enable sending the key via public channels and at the same time not compromise security.

Diffie and Hellman [2] analyze the need for public-key cryptosystems necessary for a large number of users. If a private-key cryptosystem would be in use instead, this would result in a large number of pairs ($\frac{n^2-n}{2}$) wishing to communicate privately separate from the other users. This is not reasonable for a large number of users, and Diffie and Hellmans conclusion is that a public-key cryptosystem is one solution for this.

One other later example is mentioned by Rivest, Shamir and Adleman [3] that displayed the cryptosystem using number theory. They mention that one of the important factors maintaining the security of a public-key cryptosystem is factoring of large numbers. Factoring large numbers has been a well-known problem for a long time, and for example at the time of the writing of the article, no efficient algorithm for factoring a 200-digit number had been found [3]. This is also said to be a reasonable length for n, where the decryption key consists of the positive integers d and n. n represents the product of two hidden primes p and q, and if n has a length of 200 digits, the time to factor n is calculated to take around 4 billion years. All obvious approaches are shown to have at least the same difficulty as factoring n.

One other exhibition is done by Merkle and Hellman [4] where the difficulty of the integer-packing knapsack problem is used. However, it is cautioned to not be proven secure but explains the method for constructing trap-door knapsacks, which includes forming a subset of all knapsacks and therefore the difficulty lies in the difficulty of the knapsacks [4]. On the other hand, McEliece proposes a public key cryptosystem that is based on theory of algebraic codes and uses the fact that a fast decoding algorithm exists for a general Goppa code and not general linear code.

2.1 What are Goppa Codes?

Goppa codes, named after their inventor Valery Goppa, are a class of linear error-correcting codes used in cryptography and information theory. A linear error-correcting code is a code in which any linear combination of codewords results in a new codeword [5]. These codes are particularly notable for their use in the cryptosystem described in this paper, which is one of the earliest public-key cryptosystems. They are in the class of linear error-correcting codes which are NP-complete to apply general decoding.

A Goppa code is defined over a finite field $\mathbb{F}_q$ where q is a prime power. The general execution involves two main components:

1. Goppa Polynomial: A polynomial $g(x)$ of degree t over $\mathbb{F}_q$ that is irreducible over $\mathbb{F}_q$.

2. Support Set: A set L = $\{a1, a2, a3, ..an\}$ of n distinct elements from

$\mathbb{F}_q$ where $n \leq q$.

The Goppa Code is then constructed as follows:

1. Parity Check Matrix: An $mt \times n$ matrix H (where m is the order of $\mathbb{F}_q$) is created where each column is $[1/g(a_j)$, $a_j/g(a_j)$, $a_j^2/g(a_j)$,..., $a_j^{t-1}/g(a_j)]^T$
2. Code Generation: The Goppa Code is found through the null space of H. Thus, all of the vectors v that satisfy $Hv = 0$ account for The Goppa Code [6].

3 Description of the System

The paper *A Public-Key Cryptosystem Based on Algebraic Coding Theory* describes how algebraic coding theory can be applied to public-key cryptosystems. The proposed encryption and decryption algorithms are easy to implement using digital logic and provide fast communication rates. The paper claims communication rates of at least $10^6 \frac{bits}{seconds}$ to have been feasible at the time of publishing.

3.1 Before encryption and decryption

To use the encryption and decryption algorithm, the designer needs different things to perform the algorithms:

1. n and t needs to be picked, where an irreducible polynomial is then randomly selected of degree t over a finite space. For this, Goppa code of length $n = 2^m$ exists where the dimension is $k \geq n - tm$
2. The designer generates a $k \times n$ generator matrix G.
3. G is scrambled into a random matrix $k \times k$ nonsingular matrix S and a random $n \times n$ permutation matrix P. From this, the matrix to be sent over the channel is $G' = SGP$.

3.2 The Encryption And Decryption Algorithm

To encrypt the data, we firstly divide it into k-bit blocks. If u is one block, the vector x is transmitted and received on the other end. x is calculated by this formula:

$$x = uG' + z \tag{1}$$

G' is the public generator matrix and z is a locally generatod vector of length n and weight t.

To decrypt the data, we compute x':

$$x' = xP^{-1} \tag{2}$$

P^{-1} is the inverse of permutation matrix P and x' is used as codeword in the chosen goppa code. The final to steps are to calculate $uS = u'$ with Patterson's algorithm and compute:

$$u = u'S^{-1} \tag{3}$$

4 Security Analysis

The security of the system is determined to be dependent on the attacker knowing the publicly available generator matrix G' and stopping the distribution of x, to use x and G' in establishing u. The two main approaches are

(1) Recover G from G', and then use Patterson's algorithm to find u.

(2) Recover u from x entirely without finding G.

The first approach is deemed hopeless by the paper, as the possible values for G, P and S are too many given n and t are large enough. The system designer thus need to choose n and t so that this approach is infeasible.

The essential problem of the second approach is to decode an arbitrary (n, k) linear code in the presence of t errors. Decoding linear codes is NP-complete [7], meaning this attack should too be infeasible as long as n and k are large enough.

Another crucial security aspect is that the decryption algorithm described cannot be employed as an encryption algorithm to generating unforgeable signatures. Unless the input to the decryption algorithm is a vector within Hamming distance of some codeword (i.e. the vector is similar to that codeword, with only a few bit positions differing), the decryption algorithm is highly unlikely to produce any output at all. Because only a small fraction of the 2^n possible binary vectors of length n have this property, it is even less feasible to use the decryption algorithm as an encryption algorithm.

4.1 Proof of Security

The following proof is given in the paper for the security of the public-key cryptosystem: Assume the length $n = 2^{10}$ and weight $t = 50$ for the local random vector z. The dimension is calculated by:

$$n = 2^m \tag{4}$$

$$k \leq n - tm \tag{5}$$

$$k \leq 1024 - 50 * 10 = 524 \tag{6}$$

McEliece proposes two different intuitive approaches, a brute-force method and a more effective approach assuming the selected coordinates are not erronous.

1. **Brute-Force By Comparing *x* to Each Codeword:** This approach involves comparing the intercepted data vector x to every possible codeword in the binary space. The work factor for this method is calculated as 2^{524}, which is derived from the dimension $k = 524$ of the code. Due to the vast number of computations required (2^{524} comparisons), this method is impractical with current technological capabilities.

2. **Brute-Force Based on Coset Leader:** In this refined brute-force method, the attacker identifies a coset leader for each possible coset and compares x against these leaders. A coset leader is defined as the shortest vector in a coset, and each coset corresponds to a set of vectors that differ from a particular codeword by some error vector. Although this approach reduces the number of comparisons, the work factor remains prohibitively high, rendering the attack unfeasible. The work factor for this is the size of the defined space divided by the size of the code:

$$\frac{2^{1024}}{2^{524}} = 2^{500} \tag{7}$$

3. **Probability-Based Approach:** An alternative strategy is to select k coordinates from n randomly, hoping that none are erroneous, and then

attempting to solve for the original data vector u. The probability of selecting k error-free coordinates is extremely low, especially for large values of n and t. The work factor for this approach is approximately 2^{65}, which is still beyond practical computational reach.

5 Conclusion

In conclusion, the McEliece cryptosystem is one of the milestones of advancement in the field of cryptography, providing high performance and robust security. One of its most notable features is its capacity to support exceptionally high data transmission rates, potentially exceeding 1 million bits per second. This capability renders it an strong solution option for high-speed, multi-user communication networks, where efficiency and speed are paramount. The cryptosystem's design, relying on straightforward digital logic, provides ease of application while maintaining high throughput, an important requisite in modern communication infrastructures.

In terms of security, the McEliece cryptosystem stands out due to its effective utilization of algebraic coding theory, particularly the complexities involved in decoding linear codes. The system's strength is rooted in the computational difficulty of decoding processes and the vast array of possibilities in configuring the generator matrix. These aspects collectively contribute to the system's high level of security, making it a reliable option for secure communications. The McEliece cryptosystem, therefore, emerges as not only a practical method for public-key encryption but also as a promising and secure solution tailored for the demands of high-speed network environments. Its unique approach to encryption, balancing speed and security, positions it as a forward-looking technology in the field of cryptography.

6 Follow-up work

Three examples of follow-up work are cryptoanalysis based on polar codes [8], a note with suggestions by Yang and Liang [9] and one other by Adams and Meijer [10].

References

[1] Robert J. McEliece. “A Public-Key Cryptosystem Based on Algebraic Coding Theory”. In: *Dsn Progress Report* 42.44 (1978), pp. 114–116.

[2] Whitfield Diffie and Martin E. Hellman. “New Directions in Cryptography”. In: *IEEE Transactions on Information Theory* 22.6 (1976), pp. 644–654.

[3] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. “A Method for Obtaining Digital Signatures and Public-Key Cryptosystems”. In: *Communications of the ACM* 21.2 (1978), pp. 120–126. DOI: 10.1145/359340.359342.

[4] Ralph C Merkle and Martin E Hellman. “Hiding Information and Signatures in Trapdoor Knapsacks”. In: *IEEE Transactions on Information Theory* 24.5 (1978), pp. 525–530.

[5] *Linear code – Wikipedia*. [Online; accessed 26. Feb. 2024]. URL: https://en.wikipedia.org/wiki/Linear_code.

[6] Tanja Lange. “Goppa codes: definition and usage”. Slide presentation, available online: https://hyperelliptic.org/tanja/teaching/pqcrypto21/slides/cbc-mm-3.pdf. 2021.

[7] E. Berlekamp, R. McEliece, and H. van Tilborg. “On the inherent intractability of certain coding problems (Corresp.)” In: *IEEE Transactions on Information Theory* 24.3 (1978), pp. 384–386. DOI: 10.1109/TIT.1978.1055873.

[8] Magali Bardet et al. “Cryptanalysis of the McEliece Public Key Cryptosystem based on Polar Codes”. In: *Post-Quantum Cryptography - PQCrypto 2016*. Fukuoka, Japan, Feb. 2016. DOI: 10.1007/978-3-319-29360-8_9. URL: https://hal.archives-ouvertes.fr/hal-01240856.

[9] Li Yang and Min Liang. “A Note on Quantum McEliece Public-Key Cryptosystem”. In: (Apr. 2013). arXiv:1212.0725v4 [quant-ph] 20 Apr 2013. arXiv: 1212.0725v4 [quant-ph]. URL: https://arxiv.org/abs/1212.0725v4.

[10] Carlisle M. Adams and Henk Meijer. *Security-Related Comments Regarding McEliece’s Public-Key Cryptosystem*. Tech. rep. Kingston, Canada: Department of Computing and Information Science, Queen’s University, Aug. 1986.