Multidisciplinary Perspectives in Cryptology and Information Security

Sattar B. Sadkhan Al Maliky
University of Babylon, Iraq

Nidaa A. Abbas
University of Babylon, Iraq

A volume in the Advances in Information Security, Privacy, and Ethics (AISPE) Book Series

Information Science REFERENCE
An Imprint of IGI Global

Managing Director: Lindsay Johnston
Production Editor: Jennifer Yoder
Development Editor: Erin O'Dea
Acquisitions Editor: Kayla Wolfe
Typesetter: John Crodian
Cover Design: Jason Mull

Published in the United States of America by
Information Science Reference (an imprint of IGI Global)
701 E. Chocolate Avenue
Hershey PA 17033
Tel: 717-533-8845
Fax: 717-533-8661
E-mail: cust@igi-global.com
Web site: http://www.igi-global.com

Library of Congress Cataloging-in-Publication Data

Multidisciplinary perspectives in cryptology and information security / Sattar B. Sadkhan Al Maliky and Nidaa A. Abbas, editors.
pages cm
Includes bibliographical references and index.
ISBN 978-1-4666-5808-0 (hardcover) -- ISBN 978-1-4666-5809-7 (ebook) -- ISBN 978-1-4666-5811-0 (print & perpetual access) 1. Data encryption (Computer science) 2. Wireless communication systems--Security measures. 3. Computer networks--Security measures. I. Al Maliky, Sattar B. Sadkhan, 1954- editor of compilation. II. Abbas, Nidaa A. editor of compilation.
QA76.9.A25M846 2014
005.8'2--dc23
2014011686

This book is published in the IGI Global book series Advances in Information Security, Privacy, and Ethics (AISPE) (ISSN: 1948-9730; eISSN: 1948-9749)

British Cataloguing in Publication Data
A Cataloguing in Publication record for this book is available from the British Library.

Chapter 5
Cryptography Based on Error Correcting Codes: A Survey

Marek Repka
Slovak University of Technology, Slovak Republic

Pierre-Louis Cayrel
Université de Saint-Etienne, France

ABSTRACT

Breaking contemporary cryptographic algorithms using any binary computer has at least sub-exponential complexity. However, if a quantum computer was used effectively, then our asymmetric cryptography would not be secure anymore. Since the code-based cryptography (cryptography based on error-correcting codes) relies on different problems, it is not as threatened as, for example, RSA or ECC. Recent years have been crucial in the progress of cryptography based on error-correcting codes. In contrast to the number-theoretic problems typically used in cryptography nowadays, certain instances of the underlying problems of code-based cryptography remain unbroken even employing quantum cryptanalysis. Thus, some code-based cryptography constructions belong to the post-quantum cryptography, especially cryptosystems based on binary irreducible Goppa codes. Many attempts to replace this underlying code in order to reduce the key size already have been proposed. Unfortunately, almost all of them have been broken. For instance, just a while ago, Reed Muller, Generalized Reed-Solomon Codes, and Convolutional codes were broken. Against some rank metric codes, a new attack was introduced. On the other hand, two prospective countermeasures in order to hide the exploitable code structure of the broken codes were fashioned. However, only the choice of binary irreducible Goppa codes remains secure in the post-quantum sense. This chapter surveys the more recent developments in code-based cryptography as well as implementations and side channel attacks. This work also recalls briefly the basic ideas, and provides a roadmap to readers.

DOI: 10.4018/978-1-4666-5808-0.ch005

INTRODUCTION

Present asymmetric cryptography is mainly based on discrete logarithm problem, such as cryptosystems like Diffie-Hellman key exchange protocol, DSA, or ElGamal. Elliptic Curve Cryptography (ECC) relies on the hardness of discrete logarithm defined over elliptic curves (ECDLP). Another problem is the integer factorization problem that RSA is based on. Note that the first purpose, the elliptic curves were used in cryptography, was integer factorization. In solving the discrete logarithm or factorization problem, fast progress has been made. For integer factorization, the general number field sieves are used, and for solving the discrete logarithm, the index calculus algorithm can be employed. Even the binary computation power has grown as Moore's law states. We have new integrated circuit technologies, and special devices for cryptanalysis. We have clusters, grids and clouds today. But we still believe that the complexity of present asymmetric cryptography is at least sub-exponential. Since cryptography based on error-correcting codes relies on different problems as the discrete logarithm or integer factorization, it is not threaten by breaking those problems. Furthermore, a more serious problem is that the threat of an effective use of quantum computer has arisen significantly. Clearly, if an adversary is able to use a quantum computer effectively, almost all the asymmetric cryptography algorithms are threatened (Shor, 1997). The Shor's quantum algorithm (Shor, 1997) complexity is analyzed and improved in (Zalka, 1998). Regarding this quantum algorithm, the discrete logarithm problem as well as the integer factorization problem has polynomial complexity, even on a quantum computer. The case of ECDLP quantum cryptanalysis is investigated more deeply in (Proos, & Zalka, 2003). In certain cases of the error-correcting code-based cryptography, there is no better than a sub-exponential quantum attack (Bernstein, 2010).

As mentioned above, the code-based cryptography is based on different problems as usually used today. This, in certain cases, gives the code-based cryptography a feature that is called quantum attack resistance. Quantum attack resistance means that the problem a cryptographic primitive is based on is NP-complete, thus at least sub-exponential, to solve on binary and quantum computer. Such cryptography is called post-quantum cryptography. Note that another possibility how post-quantum cryptography can be constructed is to use lattices, multivariate quadratic equations, or hash functions. More about post-quantum cryptography can be found in the book of Bernstein, Buchmann, and Dahmen (2008).

Yes, post-quantum cryptography is unbreakable today. But, despite the post-quantum cryptography being very resistant to known attacks, problems can arise when such strong cryptographic algorithms, like McEliece PKC (McEliece, 1978) or Niederreiter PKC (Niederreiter, 1986), are implemented in real devices, and post-quantum cryptography is not an exception. By this, we are pointing out the Side-Channel Attacks, which we summarize in the Section Cryptanalysis.

In this work, we are focused on the progress in the last three years particularly, thus we refer a reader that is interested in the earlier work in this field to the (Overbeck, & Sendrier, 2008) or Cayrel et al. (2011) surveys.

This chapter is sectioned as follows. Construction of the code-based cryptography is surveyed in Section Code-based cryptography survey. In this section we mention the underlying problems used in this kind of cryptography and their variants, we list cryptographic primitives and attempts to replace the fundamental error-correcting code, the binary irreducible Goppa code, in order to reduce key size. Many of the attempts to replace the code in order to reduce the key size have not

been successful as we will see, and thus the key size still remains as an issue. In Section Post-quantum cryptography based on error-correcting codes, we list cryptosystems and their constructions that have not been broken yet, and thus belong to the post-quantum cryptography. The following section, the Section Implementations, provides an overview of implementations published. In this section, we will see that post-quantum cryptography can be better performing than RSA, or even ECC. The Section Cryptanalysis consists of a survey of recent progress in side channel attacks, structural attacks, and generic attacks like information set decoding, collision decoding, or general birthday attacks. Thus, it summarizes the recent work in cryptanalysis of cryptography based on error-correcting codes. Finally, we conclude this chapter in Section Conclusions.

CODE-BASED CRYPTOGRAPHY SURVEY

In the code-based cryptography, there exist several promising candidates to public-key cryptosystem (PKC). Without a doubt, the most inspiring code-based cryptosystem is the McEliece PKC (McEliece, 1978). The McEliece PKC has never been the subject of as much attention as RSA (1977), mostly because of the relatively large size of its private and public-keys. Everything changed when it was observed that these schemes have been untouched by quantum cryptanalysis. The original McEliece PKC has been resistant to all known attacks (even using quantum cryptanalysis) for more than 35 years. Its resistance roots from the general decoding problem (Berlekamp, McEliece, & van Tilborg, 1978). Note that it is resistant to all known attacks except for the side channel attacks (Section Cryptanalysis), but what cryptographic scheme is not. Many contributions have been made in the last few years in this field. Significant part of them dealt with the key size reduction by altering the underlying error-correcting code. Almost all these attempts failed.

Inspired by the McEliece PKC, another very promising possibility for a public-key cryptosystem is the Niederreiter PKC proposed by Niederreiter (1986). This design is based on equivalent problem as the one that McEliece PKC is based on, namely the syndrome decoding problem. Also the Niederreiter PKC has not escaped the key size issue. But, now it is experimentally verified that while McEliece PKC and Niederreiter PKC require bigger keys, an implementation of McEliece and Niederreiter PKC can perform better than a RSA or even ECC implementation (Biswas, & Sendrier, 2008; Heyse, & Güneysu, 2013). Moreover, regarding the key size issue, in these times, a McEliece PKC implementation on a smart card is not a problem (Strenzke, 2010). We list more details in Section Implementations.

As we will see below, using code-based cryptography, one can construct cryptographic primitives like public-key encryption, signatures, identification, hash functions, stream ciphers, or pseudo-randomness. Their construction is based on special instances of the two crucial NP-complete decision problems in coding theory we list in the following subsection. The following two instances are used in particular. They are equivalent in terms of complexity:

1. General decoding problem that is used in two variants, namely general decoding problem over binary and q-ary finite fields.
2. Syndrome decoding problem used in three variants, namely syndrome decoding problem over binary and q-ary finite fields, and syndrome decoding problem in rank metric.

The general decoding problem relies on the difficulty of correcting and decoding a word given the code generator matrix, or given the parity check matrix equivalently. And the syndrome decoding problem relies on the hardness of finding the error

vector that corresponds to a given parity check matrix and a given syndrome.

The Two Crucial NP-Complete Problems Used in Code-Based Cryptography

In the following text, we recall the two well-known decision NP-complete problems that error-correcting code-based cryptography is based on. All other problems used in code-based cryptography are derived from them, and thus they can be reduced to them (Berlekamp, McEliece, & Tilborg, 1978). A definition list of the derived problems can be found, for example, in (Overbeck, & Sendrier, 2008).

We state the two NP-complete problems here as problems stated in NP-completeness theory. There is an input that is encoded into binary string and then fed into deterministic Touring machine. The machine outputs 1 if the input has the desired property, and it outputs 0 if otherwise. In certain cases of linear error-correcting codes, there is no such a machine that would produce the correct answer in polynomial time even it was a quantum computer. The following two decision problems are NP-complete (Berlekamp, McEliece, & van Tilborg, 1978).

- **The Coset Weights Decision NP-complete Problem:**
  - **Input:** A binary matrix H, a binary vector **s**, and a nonnegative integer w.
  - **Property:** There exists a vector e of hamming weight less then w such that $s = eH$.
- **The Subspace Weights Decision NP-complete Problem:**
  - **Input:** A binary matrix H, and a non-negative integer w.
  - **Property:** There exists a vector x of hamming weight w such that $xH = 0$.

The first decision NP-complete problem listed here can be viewed as following minimization problem. Let us have a linear code of length n and dimension k. We received word y. Its syndrome s is computed as $s = yH$, where H is the $n \times \left(n - k\right)$ parity check matrix of the code. There is a solution e of the minimum weight that gives the same syndrome, i.e. $s = eH$. The solution e is the best approximation of the error vector that has been added to the transmitted code word x trough a binary symmetric channel.

The second NP-complete decision problem stated here can be viewed as decision problem whether a linear code contains a code word of the given weight.

List of Cryptographic Primitives

A brief list of the last proposals or improvements of cryptographic primitives based on error-correcting codes follows. For a more detailed list, we refer a reader to the survey published by Cayrel (2011).

Encryption: For the encryption purpose, we have the McEliece PKC, or Niederreiter PKC (Section Post-quantum cryptography based on error-correcting codes). An appropriate choice of parameters for them is listed in the paper (Bernstein, Lange, & Peters, 2008). We should also not forget about the Indistinguishability under Adaptive Chosen Ciphertext Attack (IND-CCA2) conversions necessity (see below). Note, encryption cryptosystem presented in (Lu et al., 2011) is IND-CCA2-secure naturally.

Since the key size issue, several attempts to replace the underlying code have been designed. Unfortunately, many attempts to change the code the cryptosystems are based on were not successful. So we have to be very careful in modifications of the McEliece/Niederreiter PKC. Also in the Niederreiter PKC the Goppa codes designed for the McEliece PKC should be used. In order to

hide the exploitable structure of a code, one can try to use one of the countermeasures listed below.

Identification: Recently, Cayrel, Véron, and Alaoui (2011) presented an identification scheme using q-ary codes instead of binary codes. Based on maximum-rank-distance codes, a new proposal has been introduced (Gaborit, Schrek, & Zémor, 2011).

Signature: Barreto, Misoczki, and Simplício (2011) developed the syndrome-based one-time signature scheme (BMS-OTS). An improved version of a threshold ring signature can be found in (Cayrel et al., 2012), a one-time signature scheme is presented in (Gaborit, & Schrek, 2012), and a signcryption scheme is proposed in (Mathew, Vasant, & Rangan, 2012).

Pseudo-randomness: Fischer, and Stern (1996) presented the first pseudo-random generator based on error-correcting codes. Security of the generator is conditioned by the hardness of the syndrome decoding problem for random binary linear codes. Hence, the generator is based on the fact that the greater the weight of error vectors, the exponentially greater the number of words having the same syndrome. Regarding the code length and the dimension of a linear error-correcting code, the weight of error vectors is chosen close to the Gilbert-Varshamov bound (Gilbert, 1952; Varshamov, 1957). The sampling algorithm originally proposed for this generator can be found in the (Fischer, & Stern, 1996). Another very good choices are mentioned in (Biswas, & Sendrier, 2008; Heyse, 2010).

Digest: Bernstein et al. (2011) proposed RFSB (which stands for Really Fast Syndrome-Based Hashing). RFSB is based on random functions, and uses the AES algorithm.

Stream ciphers: Using the ideas from RFSB, Meziani, Hoffmann, and Cayrel (2012) improved the code-based stream cipher SYND.

On the Underlying Error-Correcting Code

Since the key size of the original McEliece PKC is large in comparison with RSA or ECC, there has been proposed many attempt to replace the irreducible binary Goppa codes. Unfortunately, almost all of them were not successful. In this subsection, only unbroken error-correcting codes are listed. The full list of codes proposed to be used in code-based cryptography, and whether the proposed code was broken or not, can be found in the Table 1.

Binary irreducible Goppa codes with maximal length: The concept to use binary irreducible Goppa codes with maximal length routes from McEliece (1978). This choice of error-correcting code still remains unbroken even using quantum

Table 1. State of error-correcting codes survival in code-based cryptography

Underlying Error-Correcting Code	Attack
Binary irreducible Goppa codes	No attack
Goppa codes over q-ary finite fields ($q \geq 31$)	No attack
Moderate density parity-check codes	No attack
Convolutional codes	(Landais, & Tillich, 2013)
Generalized Reed-Solomon Codes and sub-codes	(Couvreur et al., 2013; Gauthier, Otmani, & Tillich 2012)
Goppa codes over q-ary finite fields ($2 < q < 31$)	(Peters, 2010)
Maximum-rank-distance codes	(Gaborit, Ruatta, & Schrek, 2013)
Reed-Muller codes	(Chizhov, & Borodin, 2013)
Quasi-Cyclic Alternant, Quasic-Dyadic Goppa, BCH, Low density parity-check codes	(Faugère et al., 2010)
Algebraic geometric codes in case of low genus hyperelliptic curves	(Faure, & Minder, 2008)
Generalized Srivastava codes	(Sidelnikov, Shestakov, 1992)

cryptanalysis. Thus, in construction of post-quantum error-correcting code-based cryptography, random instances of this error-correcting code are used. We define this codes in Section Post-quantum cryptography based on error-correcting codes.

Goppa Codes over q-ary finite fields: Bernstein, Lange, and Peters (2010) demonstrated usage of Goppa codes defined over fields that are characteristic of power of an odd prime (q-ary finite fields). This cryptosystem is called Wild McEliece. Bernstein, Lange, and Peters (2010) claimed that, using smaller keys, they achieve the same security level as the original McEliece PKC. Their proposal had just one weakness. The pool of Goppa polynomials is not as big as in the original McEliece PKC case. This vulnerability can be misused by the support splitting algorithm (Sendrier, 2000). In order to eliminate this vulnerability, Bernstein, Lange, and Peters (2011) introduced the Wild McEliece Incognito. In this PKC, the Goppa polynomial is multiplied by a co-prime polynomial. In these codes it is possible to correct more errors than in the case of binary irreducible Goppa codes.

Moderate density parity-check codes: Since (Faugère et al., 2010), the Low-Density Parity-Check (LDPC) codes (and their Quasi-Cyclic variant) were replaced by Moderate Density Parity-Check Codes (MDPC) that has been suggested to use by Misoczki et al. (2013).

Promising Countermeasures for the Weak Error-Correcting Code Choices

Baldi, Bodrato, and Chiaraluce, (2008) showed that if the secret permutation matrix is changed to a dense transformation matrix increasing the density of the public parity check matrix, the attacks against cryptosystems using QC, QD, Generalized Srivastava and its sub-codes, and also LDPC codes become significantly more complex. This idea has been improved and generalized into a non-binary case in the work (Baldi et al., 2011).

Another very promising modification of the McEliece PKC seems to be the work (Gueye, & Mboup, 2013). They modified the key generation algorithm in a way that a random matrix is connected to the private-code generator matrix from the right. Then the new matrix is hidden multiplying it by a secret random dense scramble matrix, and by a secret random permutation matrix, as was originally proposed (McEliece, 1978). The decoding algorithm is also modified. After the secret permutation is removed from the cipher-text, only the first number of bits (equal to the code length) are considered in the following process, which is as originally proposed in (McEliece, 1978). This modification is presented on Reed-Muller codes that were broken (Chizhov, & Borodin, 2013).

IND-CCA2-Secure Conversions

Clearly, the original McEliece/ Niederreiter PKC proposal is vulnerable to the adaptive chosen-ciphertext attack (CCA2). Possible CCA2-secure conversions can be found, for instance, in (Kobara & Imai, 2001; Dowsley, Müller-Quade, & Nascimento, 2009; Dottling et al., 2012). An efficient CCA2-secure variant of the McEliece PKC in the standard model can be found in (Rastaghi, 2013). The indistinguishability under adaptive chosen cipher-text attack (IND-CCA2) is addressed by Persichetti (2012). Mathew et al. (2012) proposed a new and efficient IND-CCA2-secure conversion of the Niederreiter PKC.

POST-QUANTUM CRYPTOGRAPHY BASED ON ERROR-CORRECTING CODES

Many attempts to change the underling error-correcting code, or to modify the original McEliece (1978) or Niederreiter (1986) proposal, were not successful, and those proposals that have not been broken yet can be broken in the close future. Only the original McEliece proposal (to use the

binary irreducible Goppa codes with the maximal length) is credible. This original proposal has not been broken since 1978. The next credible post-quantum code-based public-key cryptosystem is the Niederreiter PKC, but only when the binary irreducible Goppa codes with maximal length are used. For better understanding, let us recall those two cryptosystems, and the underlying error-correcting code as well as the Patterson's algebraic decoding algorithm (Patterson, 1975).

Binary Irreducible Goppa Codes with Maximal Length

Goppa codes were introduced by Goppa (1970). In the original McEliece PKC proposal (McEliece, 1978), random binary irreducible Goppa codes are used. However, in PKCs like McEliece and Niederreiter, not only binary irreducible Goppa codes can be used. Unfortunately, cryptanalytic community identified most of them to be less secure or less efficient than the binary irreducible Goppa codes, as we mentioned in Section Code-based cryptography survey. Thus, Goppa codes (Goppa, 1970) play a special role in the code-based cryptography, especially in construction of the post-quantum cryptography based on error-correcting codes. Let us sketch out definition of binary irreducible Goppa codes designed for McEliece PKC.

Let us have a finite field characteristic of two. Irreducible Goppa polynomial $g(X)$ is a monic polynomial that is irreducible over the finite field. Code support Λ is an ordered set of λ_i, or a vector of all the elements of the finite field, that are not zeros of the Goppa polynomial. Since the Goppa polynomial is irreducible, we have all the field elements in the code support. The code support elements are distinct of course. A binary irreducible Goppa code is a linear code represented by the code support and the irreducible Goppa polynomial. Obviously, a binary vector c length of n, the code length, is in the code if its syndrome polynomial $S(X)$ is congruent to zero modulo the irreducible Goppa polynomial.

$$\Gamma\left(\Lambda, \mathrm{g}\right) = \left\{\mathrm{c} \in F_{2^n} \mid S_c\left(X\right) \equiv 0 \bmod g\left(X\right)\right\}, \tag{1}$$

$$S_c\left(X\right) = \sum_{i=0}^{n-1} \frac{c_i}{X - l_i}. \tag{2}$$

This binary irreducible Goppa code has length equal to the number of the code support elements, i.e. the number of the field elements in this case. The dimension of this code is equal to the number of the field elements minus the degree of the irreducible Goppa polynomial multiplied by the degree of the irreducible polynomial used to create the finite field. Finally, the minimum distance of the code is at least two times the degree of the irreducible Goppa polynomial plus one.

Goppa codes are a subclass of Alternant codes (Helgert, 1974). This knowledge can be used in order to reduce required key size. But nobody has been successful yet (Section Code-based cryptography survey). On the other hand, the same knowledge can be misused in order to design a code distinguisher (Faugère et al., 2013). Such a Goppa code distinguisher, which is able to distinguish a Goppa code generator matrix from a random matrix, can threat McEliece PKCs.

Patterson's Algebraic Decoding Algorithm

In the Patterson's algebraic decoding algorithm (Patterson, 1975), the syndrome of an error vector has to be determined first. It can be determined using the secret parity check matrix, or simply by evaluating the syndrome polynomial $S(X)$ modulo the irreducible Goppa polynomial $g(X)$.

$$S_e\left(X\right) \equiv \sum_{i=0}^{n-1} \frac{y_i}{X - \lambda_i} \bmod g\left(X\right). \tag{3}$$

In the syndrome polynomial, y_i is the *i*-th bit of a being corrected word, λ_i is the *i*-th element of the code support, and *n* is the code length as well as the number of code support elements. When the error syndrome is determined, the next main step is to compute the corresponding error-locator polynomial.

An error-locator polynomial $\sigma(X)$ is a polynomial over the finite field the code is defined. Positions of zeroes of the error-locator polynomial in the code support gives error bit positions in the word that is being corrected.

$$\sigma_e\left(X\right)=\prod_{i=0}^{n-1}\left(X-\lambda_i\right)^{e_i}. \tag{4}$$

The syndrome polynomial is congruent to the fraction of the error-locator polynomial's first derivative and the error-locator polynomial, modulo the irreducible Goppa polynomial.

$$S\left(X\right)\equiv\frac{\sigma'\left(X\right)}{\sigma\left(X\right)}\bmod g\left(X\right). \tag{5}$$

Since a binary irreducible Goppa code is being corrected, the first derivative of the error-locator polynomial consists only of all the even terms, i.e. the error-locator polynomial can be split into squares and non-squares, see below where

$$\beta^2\left(X\right)=\sigma'\left(X\right). \tag{6}$$

$$\sigma\left(X\right)=\alpha^2\left(X\right)+X\beta^2\left(X\right). \tag{7}$$

Therefore, the error-locator polynomial multiplied by the syndrome polynomial is congruent to the first derivative of the error-locator polynomial, modulo the irreducible Goppa polynomial. This equation can be rewritten as the square root of the first derivative of the error-locator polynomial times the square root of (*X* plus the inversion of the syndrome) congruent to the square root of non-square terms in the error-locator polynomial, modulo the irreducible Goppa polynomial. This equation is called Key equation.

$$T\left(X\right)\equiv X+S^{-1}\left(X\right)\ mod\ g\left(X\right). \tag{8}$$

$$b\left(X\right)\sqrt{T\left(X\right)}\equiv\alpha\left(X\right)\ mod\ g\left(X\right) \tag{9}$$

The key equation is solved by applying the Extended Euclidean Algorithm that stops when the

$$\deg\left(\alpha_k\left(X\right)\right)\le\left\lfloor\frac{\deg\left(g\left(X\right)\right)+1}{2}-1\right\rfloor, \tag{10}$$

where *k* is iteration number of the Extended Euclidean Algorithm. Once the square root of the error-locator polynomial's first derivative and the square root of non-square terms of the error-locator polynomial are revealed, the error-locator polynomial is computed by squaring them, and after the multiplication of the error-locator polynomial's first derivative by *X*, by adding them.

In this case, the case of McEliece/Niederreiter PKC, the resulting error-locator polynomial has degree equal to the degree of the irreducible Goppa polynomial. This is the maximum number of errors the Patterson's algebraic decoding algorithm is capable to correct, and also the maximum number of errors that can be corrected in any binary irreducible Goppa code.

Finally, in order to determine the error vector, the roots of the error-locator polynomial have to be found. For this purpose, one can simply evaluate the obtained error-locator polynomial over the code support or use a factorization method like the Berlekamp trace algorithm (Berlekamp, 1971), or Chien method (Chien, 1964), or any other. Efficiency of some factorization methods that can be used here is discussed in (Strenzke,

2011). Finally, indexes of the code support elements that are zeros of the error-locator polynomial give indexes of error bits in the received word.

As an alternative to the Patterson's algebraic decoding algorithm (Patterson, 1975), one can use decoding algorithm described in (Sugiyama et al., 1976) which is very similar to the Patterson's one, or recently published List decoding for binary Goppa codes that was presented by Bernstein (2011).

The McEliece PKC

The McEliece PKC (McEliece, 1978) is based on the general decoding problem. More precisely, it is based on the problem how to find the secret code which is permutation equivalent to the public one. It still remains unbroken in this original design where the binary irreducible Goppa codes with maximal length are used.

A private key K_{priv} consists of a secret random binary irreducible Goppa code (Goppa, 1970) with maximal length n, and dimension k, as we defined above. The corresponding public-key K_{pub} is derived from the private one so that the generator $k \times n$ matrix G_{priv} of the secret code is multiplied by a random dense invertible scramble $k \times k$ matrix S and a random permutation $n \times n$ matrix P respectively, which are also secret. Those random matrices are then the next part of the private key.

$$G_{pub} = SG_{priv}P\,. \tag{11}$$

The Generator matrix G_{pub} of the public binary irreducible Goppa code together with t that is the degree of the irreducible Goppa polynomial form the corresponding public-key. It is clear that the public generator matrix generates a permutation equivalent code to the private one. Therefore,

$$K_{priv} = \left(\Gamma\left(\Lambda, g\right), S, P\right), \tag{12}$$

where the code support is randomly ordered, and g is a random irreducible Goppa polynomial, and

$$K_{pub} = \left(G_{pub}, t\right). \tag{13}$$

The encryption algorithm is very simple. Given a public-key, one message block m is encoded to the corresponding code word. Then a random error e hamming weight of the degree of the irreducible Goppa polynomial is added to the code word. The result of these several bitwise exclusive OR (XOR) additions and binary multiplications produces a cipher-text y.

$$y = mG_{pub} + e\,. \tag{14}$$

At start of the decryption algorithm, the secret permutation matrix is removed— the public generator matrix generates a permutation code equivalent to the secret one. Subsequently, the Patterson's algebraic decoding algorithm (Patterson, 1975) is applied in order to remove the error. In the time the error vector is removed, only the information coordinates are read from the code word, and finally the secret invertible scrambling matrix is removed. As a result the plain-text is obtained.

$$m' = \mathrm{Patterson}(yP^{-1})\,, \tag{15}$$

$$m = \mathrm{getInformationCoords}(m')S^{-1}\,. \tag{16}$$

Note, the IND-CCA2 conversions are mentioned in the Section Code-based cryptography survey.

Niederreiter PKC Using Binary Irreducible Goppa Codes with Maximal Length

From a security point of view, this cryptosystem is equivalent to the McEliece PKC. It is based on the syndrome decoding problem. From the performance perspective this scheme should be better performing than the McEliece PKC.

The main difference is that instead of a generator matrix, the Niederreiter PKC uses a party check matrix only. A block of a plaintext is mapped to an error vector of desired weight. The corresponding cipher-text is then the syndrome of the error vector. It is clear that the mapping φ has to be a bijective function that is easy to compute, invert, and implement. Here are several proposals for the mapping function (Fischer, & Stern, 1996; Biswas, & Sendrier, 2008; Heyse, 2010).

In the original Niederreiter PKC proposal (Niederreiter, 1986), any linear error-correcting code can be used. However, due to the fact that most of them were turned out to be less secure or less efficient (Section Code-based cryptography survey), it is suggested to use binary irreducible Goppa codes with maximal length. Thus, we restrict the interpretation in the following paragraph in to such a case only.

A private key K_{priv} consists of a $\left(n \times mt\right)$ parity check matrix H_{priv} of a secret random binary irreducible Goppa code $\Gamma\left(\Lambda, \mathrm{g}\right)$ with maximal length. The corresponding public-key K_{pub} is generated as the product of a $\left(n \times n\right)$ random binary permutation matrix P, the secret parity check matrix, and a $\left(mt \times mt\right)$ random dense non-singular binary matrix S, respectively, resulting in a public parity check matrix H_{pub}.

$$H_{pub} = PH_{priv}S. \tag{17}$$

The corresponding public-key K_{pub}, thus, consists of the public parity check matrix and $\boldsymbol{t}$ that is the degree of the irreducible Goppa polynomial.

$$K_{priv} = \left(\Gamma\left(\Lambda, g\right), S, P\right). \tag{18}$$

$$K_{pub} = \left(H_{pub},\ t\right). \tag{19}$$

Once a public-key is given, in order to encrypt one message block m, one has to map the message block onto an error vector of the code length and hamming weight equal to the degree of the irreducible Goppa polynomial. For the mapping the bijective function φ is used. Consequently, the product of the public parity check matrix and the error vector obtained results in the cipher-text that is basically the syndrome.

$$y = \varphi\left(m\right)H_{pub}. \tag{20}$$

The cipher-text can be decrypted using the corresponding private key only. First the random dense non-singular binary matrix is removed. Then, the Patterson's algebraic decoding algorithm (Patterson, 1975) is applied. Now the random binary permutation matrix is removed. And finally, the inverse mapping is performed in order to reveal the message. For an IND-CCA2 conversion consult the Section Code-based cryptography survey.

$$\mathrm{m} = \varphi^{-1}\left(\mathrm{Patterson}(yS^{-1})P^{-1}\right). \tag{21}$$

IMPLEMENTATIONS

For more than 25 years there has not been published any paper devoted to a McEliece PKC implementation because of its large public-key size.

Encryption

McEliece: Although the key size in McEllie PKC is rather large, it is not such a problem nowadays, as can be seen, for instance, in the implementation of McEliece PKC on a smart card with an Infineon SLE 76 chip (Strenzke, 2010). Strenzke (2010) implemented two instances of McEliece PKC. One instance has the length of the code equal to 1024 with 40 errors (62 security bits), and the second instance has the code length 2048 with 50 errors (102 security bits). In order to reduce the key size of the McEliece PKC, the public-key is generated from a private one in such a way that the public generator matrix is in the systematic form. The McEliece PKC vulnerability to the adaptive chosen-ciphertext attack (CCA2) is thwarted using CCA2-secure conversion. For interest, encryption took 970ms and decryption took 690ms for the first instance, and the encryption and decryption for the second instance took 1390ms and 1060ms respectively. The QC-MDPC McEliece variant (key size 4.8KB only) was implemented by (Heyse, von Maurich & Güneysu, 2013) (Table 2, Table 3).

For embedded devices, the first McEliece PKC implementation was published by Eisenbarth et al. (2009). They implemented an instance of McEliece PKC with code length 2048 with 27 errors what corresponds to an 80 security bits, which they named MicroEliece, on an 8-bit AVR microprocessor, and on a Xilinx Spartan-3AN FPGA. They used several clever ideas –like the public code generator matrix in the systematic form, special generation method and representation of permutation and scrambling matrices and appropriate representation of the code support– in order for keys to save the memory size required. Despite the clever ideas, the FPGA was too small to implement the whole McEliece. Thus, they implemented encryption first and decryption afterwards. They achieved following results. In the case of the microcontroller, encryption process and decryption process took 450ms and 618ms respectively. For a comparison, they mentioned computation time

Table 2. Performance of McEliece PKC implemented on Microcontroller platform

Cryptosystem Implementation /Code length, Number of Errors Correcting, Security Bits/	Device	Computation Time for Encryption, and Decryption Respectively	Reference
McEliece PKC /1024, 40, 62/	Infineon SLE76CF5120P controller, 16-bit CPU @ 33 MHz	970ms, 690ms	(Strenzke, 2010)
McEliece PKC /2048, 50, 102/	Infineon SLE76CF5120P controller, 16-bit CPU @ 33 MHz	1390ms 1060ms	(Strenzke, 2010)
McEliece PKC /2048, 27, 80/	AVR ATxMega192, 8-bit CPU @ 32MHz	450ms, 618ms	(Eisenbarth et al., 2009).
QC-MDPC McEliece PKC /9600, 84, 80/	AVR ATxMega256A3, 8-bit CPU @ 32MHz	800ms, 2700ms	(Heyse, von Maurich & Güneysu, 2013).

Table 3. Performances of ECC and RSA implementations on Microcontroller platform

Cryptosystem Implementation	Device	Computation time	Reference
ECC-P160 (SECG)	ATMega128@8MHz	203ms (scaled for 32MHz)	(Eisenbarth et al., 2009).
RSA-1024	ATMega128@8MHz	20748ms (scaled for 32MHz)	(Eisenbarth et al., 2009).

for ECC-P160 (SECG) and random instances of RSA-1024 on the same platform. The computation time for the ECC-P160 (SECG) was 203ms, and the computation time for the RSA-1024 was 2748ms. On the FPGA platform, the encryption took 1.07ms and the decryption took 10.82ms while the running time for the ECC-P160 (SECG) and RSA-1024 was 5.1ms and 51ms respectively. Note that ECC is better performing here, but in the McEliece PKC, there still exist opportunities for optimization. Indeed, ECC has been studied more notoriously than McEliece PKC due to the disadvantage of the key size.

The whole McEliece PKC processor architecture for a Virtex-5 FPGA and its implementation results were published by Shoufan et al. (2009). The published architecture involves key generator, encryptor, and decryptor. Their implementation of McEliece PKC with 2048 code length correcting 50 errors spent 84% of slices and 50% of BRAMs (2700 Kb).

Finally, The QC-MDPC McEliece variant, with 4.8KB key size, was implemented on Xilinx Virtex-6 FPGA by (Heyse, von Maurich & Güneysu, 2013). The achieved results are very promising (Table 4. Table 5).

Moreover, not only hardware implementations or implementations for embedded devices have been published. The HyMES implementation in C language running under Linux was published by (Biswas & Sendrier, 2008).

Niederreiter PKC: The Niederreiter PKC was also implemented on an embedded platform (Heyse, 2010). Heyse (2010) implemented Niederreiter PKC on an 8-bit AVR ATxMega256A1 microcontroller. He used log and antilog tables for multiplication computation purpose. Niederreiter PKC with a code length of 2048 and 27 errors corresponding to an 80 security bits was implemented. The performance the implemented Niederreiter PKC achieved is comparable to the performance of ECC-P160 (SECG) and a random RSA-1024 instance. The encryption took 1.6ms and the decryption took 180ms. For ECC-P160 (SECG) the running time was 203ms, and for the random RSA-1024 instance the running time was 2748ms (Table 6).

Table 4. Performance of McEliece PKC implementations on FPGA platform

Cryptosystem Implementation /Code length, Number of Errors Correcting, Security Bits/	Device	Computation Time for Encryption, and Decryption Respectively	Reference
McEliece PKC /2048, 27, 80/	Spartan-3AN 1400 FPGA, Enc@150MHz, Dec@85Mhz	1.07ms, 10.82ms	(Eisenbarth, Güneysu, Heyse, & Paar, 2009).
McEliece PKC /2048, 50, 102/	Xilinx Virtex-5, 163MHz	0.5ms, 1.4ms	(Shoufan et al., 2009).
QC-MDPC McEliece PKC /9600, 84, 80/	Xilinx Virtex-6 Enc@351.3MHz, Dec@190.6MHz.	0.14ms, 0.86ms	(Heyse, von Maurich & Güneysu, 2013).

Table 5. Performance of ECC and RSA implementations on FPGA platform

Cryptosystem Implementation	Device	Computation time for Encryption, and Decryption Respectively	Reference
ECC-P160 (SECG)	Spartan-3 1000-4	5.1ms	(Eisenbarth et al., 2009).
ECC-K163	Virtex-II	0,0358ms	(Heyse, & Güneysu, 2012)
RSA-1024	Spartan-3E 1500-5	51ms	(Eisenbarth et al., 2009).

Table 6. Performances of Niederreiter PKC implementations

Cryptosystem Implementation / Code length, Number of Errors Correcting, Security Bits /	Platform, Device	Computation Time for Encryption, and Decryption Respectively	References
Niederreiter PKC /2048, 27, 80/	Microcontroller AVR ATxMega256A1, 8 bit CPU, 32MHz	1.6ms, 180ms	(Heyse, 2010)
Niederreiter PKC /2048, 27, 80/	FPGA, Xilinx Virtex6LX240, Enc@300MHz, Dec@250MHz	0.00066ms, 0.05878ms	(Heyse, & Güneysu, 2012)

The very promising implementation of the Niederreiter PKC was published by Heyse, and Güneysu (2012). Their implementation on Xilinx Virtex-6 FPGAs providing 80-bit security was able to run 1.5 million encryption and 17000 decryption operations per second, respectively. This result is in dimension of ECC performance. Afterwards, they optimized the implementation in (Heyse, & Güneysu, 2013).

Derived cryptosystems: On embedded platform, there have been implemented also variants considering the key size issue that is addressed by code structure. Inspirited by LDPC and MDPC codes, on embedded platform, Barreto, Misoczki, and Ruggiero (2012) implemented a variant of McEliece based on QC-LDPC and cyclosymmetric MDPC codes. Subsequently, Cayrel, Hoffmann, and Persichetti (2012) implemented a CCA2-secure McEliece variant based on QD generalized Srivastava codes.

Signature

In order to save space required for keys storing, Strenzke (2012) proposed an approach how to compute digital signature on memory-constrained devices like smart cards. The public-key is not stored on the device, but rather it is send to the device part after part. The device then computes intermediate results using the parts of the public-key it receives. In order to shrink the size of private keys, the syndrome is computed evaluating the syndrome polynomial instead of using the secret parity check matrix.

On hardware platform, Beuchat et al. (2004) published a McEliece like signature scheme implementation on Xilinx Virtex XCV300E FPGA.

Identification

Using QC codes instead of random codes, Cayrel, Gaborit, and Prouff (2008) proposed an efficient implementation of Stern's protocol on a smart-card. For the security level of 80 bits, they obtained an authentication in 6 seconds and a signature in 24 seconds without cryptographic co-processor. This is a promising result when compared to an RSA implementation which would take more than 30 seconds in a similar context.

Other Cryptographic Primitives

Further cryptographic primitives, like identification, hash function, stream ciphers, and also encryption and signature implemented in C and Java languages can be found on the web page of Cayrel (2012).

Cryptanalysis

The two main types of attacks in code-based cryptography are structural and information set decoding attacks. The structural attacks rely on the specific structure of an error-correcting code. The information set decoding attacks are generic

attacks. Further, there are code distinguishers and quantum attacks. This section starts with side channel attacks. Indeed, side channel attacks are very dangerous implementation attacks.

Side Channel Attacks

The first side channel analysis of the McEliece PKC was published by Strenzke et al. (2008). They presented a timing attack realized in the Patterson's algebraic decoding algorithm in the decryption process, a power attack on the construction of the parity check matrix during key generation, and a CACHE timing attack on the permutation of code words during decryption. The timing attack against the Patterson's algebraic decoding algorithm misuses the fact that the error-locator polynomial degree equals exactly to the number of errors in the received word (cipher-text). Thus, in order to determine the message (plain-text), one can try to request decryption of a fake cipher-text which was created XORing the true cipher-text with a vector with hamming weight one. If the flipped bit in the fake cipher-text is an error bit, then its error-locator polynomial has one less degree than the error-locator polynomial of the true cipher-text. Strenzke et al. (2008) misused this fact and they measured running time of the error-locator polynomial evaluation in the Patterson's algebraic decoding algorithm, which is linearly dependent to the degree of the error-locator polynomial. By this method, one can reveal the whole error vector. They stated that it is possible to mount this attack also against the CCA2-secure conversion like one published in (Kobara & Imai, 2001) as an example. In the following part of their work, they discuss a possibility to reveal coefficients of the irreducible Goppa polynomial assuming that the code support is known. If the code support is known, the irreducible Goppa polynomial can be revealed by analyzing the power consumption caused by the evaluation of the irreducible Goppa polynomial in the parity check matrix construction in the key generation phase. In the remaining part of the Strenzke et al. (2008) work, the CACHE timing attack against the first step of decryption is mentioned. In this attack scenario, it is assumed that the permutation matrix is stored as a look-up table in memory, and that an adversary is able to run a spy process in parallel to the decryption process. The spy process role is to prepare the CAHCE for the attack and to measure the running time of the decryption process. If it is so, it is possible to determine which part of the cipher-text was accessed. This attack works well only if the CACHE size is small enough.

The work of Strenzke et al. (2008) was worked out into a deeper timing analysis of the Patterson's algebraic decoding algorithm in (Shoufan et al., 2009). They presented that not only the error-locator polynomial evaluation leaks, but furthermore that the construction of the error-locator polynomial leaks in the same manner. The construction of the error-locator polynomial follows after the Extended Euclidean Algorithm step (used for key equation solving in the Patterson's algebraic decoding algorithm). Since the number of iterations of the Extended Euclidean Algorithm depends on the error vector hamming weight, also the Extended Euclidean Algorithm step can be misused in the same manner. These timing attacks in combination with fault injection attacks reveal just a message, and not the private key. Sensitivity of McEliece and Niederreiter like PKCs to fault injection attacks is investigated by Cayrel, and Dusart (2010).

Strenzke in the work (Strenzke, 2010) presented a different approach to the timing attack against the Patterson's algebraic decoding algorithm in the decryption process. The presented attack exploits fundamental behavior of Extended Euclidean Algorithm in the Patterson's algebraic decoding algorithm, namely the fact that if the number of errors in a received word (i.e. cipher-text) is exactly four, the number of Extended Euclidean Algorithm iterations, when solving the key equation, can be either zero or one. This obviously leads to different computation time, what allows the adversary to

determine the number of the iterations. He found that the third coefficient of the irreducible Goppa polynomial can be rewritten as a function of error positions. Therefore, if the attacker is allowed to make a number of cipher-texts with four errors (the attacker knows the error vectors), by misusing the information about the number of the iterations, the adversary can construct a list of linear equations describing the secret permutation. Afterwards, Gaussian elimination can be applied on to the equation system. In the case of code length 1024 with 27 errors, the adversary needed to generate 2,163,499 messages in average, and in the case of code length 2048 with 50 errors, 7,848,229 messages were needed to be generated. The rank of the matrix of the linear equations was 1013, and 20136 respectively.

Strenzke (2013) presented a timing attack against the syndrome inversion step in decryption process. The paper describes a timing attack revealing the secret key. It is based on his previous work (Strenzke, 2010) mentioned above. In this paper, not only cipher-text with four errors is generated, but also with one and six errors. These numbers of errors are chosen especially in order to reveal certain coefficients of the error-locator polynomial, and these coefficients reveal some information about the secret code support. Cipher-texts with one error are generated in order to reveal position of the zero element in the code support. Cipher-texts with four errors results in the list of linear equations, and cipher-texts with six errors are used to make a cubic equation system, which have to be solved respectively.

The first practical evaluation of a power analysis attack revealing the secret permutation, and the scrambling matrix, during the decryption process can be found in (Heyse, Moradi, & Paar, 2010). Heyse, Moradi, and Paar (2010) conducted and evaluated the attack against the McEliece PKC decryption algorithm implemented on the embedded device proposed by Eisenbarth et al. (2009). They assumed two main scenarios. In the first scenario, the secret permutation is removed before the Patterson's algebraic decoding algorithm. By observing a power trace of syndrome computation, an adversary can determine when additions of the parity check matrix rows are performed. In their implementation, those additions are performed sequential. Thus, a column rows are summed only if the corresponding coordinate of the cipher-text after the secret permutation removal is one. Thanks to the knowledge which columns of the parity check matrix was summed up, the secret permutation is not secret anymore. In the second scenario, in respect to the secret permutation, the parity check matrix in the syndrome computation is permuted. Next their observation is that if the Goppa polynomial is loaded, for example, at the start of the syndrome polynomial inversion, using a simple power analysis, one can predict hamming weight of all the Goppa polynomial elements. So, given the secret permutation that was revealed by the simple power analysis above, the secret scrambling matrix and the secret Goppa polynomial can be revealed. If the private generation matrix is in the systematic form, the secret scrambling matrix is revealed using public-key. The Goppa polynomial is revealed afterwards by computing the great common divisor of a two different syndromes, with high probability. In the case of the scenario two, the attack is not working. Thus, another approach is to use a simple power analysis in order to find the secret parity check matrix and reveal the secret code support from that. Each row of the parity check matrix is totally defined by a code support element, Goppa polynomial evaluated at that support element, and the Goppa polynomial coefficients. Regarding the hamming weight derived from a measured power trace, a list of candidates for parity check matrix cell is created. For each column of the parity check matrix, a code support element and its value of Goppa polynomial is chosen randomly over all the possible elements. Now, they go recursively into the rows of the actual column. At each recursion level, the corresponding coefficient of the Goppa polynomial has to be chosen randomly, and actual

cell candidate is computed. Only if this value is in the candidate list, recursion continues. If the recursion processed the last row of the parity check matrix, a code support element, its Goppa polynomial evaluation, and all the Goppa polynomial coefficients are selected. Now whether the selected Goppa polynomial evaluates in the evaluation candidate at the selected support elements has to be checked. If yes, a final candidate of the Goppa polynomial and the code support element is chosen. While the algorithm continues to search another candidates, the actual candidate is validated by the next column of the parity check matrix and next code support element. As a result of this procedure, several pairs of code support and Goppa polynomial are obtained. The correct pair decodes all the cipher-texts. The attack running time for Xeon E5345CPUs and 16GB RAM took 69 min, wherein the subject of the attack was a cryptosystem instance with code length 2048 and 27 errors.

The timing attack in combination with fault injection attack (Strenzke, 2010), which is focused on the Extended Euclidean Algorithm for solving the key equation in the Patterson's algebraic decoding algorithm, was modified by Molter et al. (2011). In order to determine whether less iterations of the Extended Euclidean Algorithm was performed, they used information about the number of peaks in a measured power trace instead of timing information.

Avanzi et al. (2011) used idea of template or profiling attacks. They extended the attack of Strenzke et al. (2008) by including a profiling phase, wherein the adversary builds a simple statistical profile of decryption computation time regarding all the possible correctable error vectors of the desired hamming weight. Thanks to this profiling phase of attack, the success of the attack has been improved significantly. Furthermore, Avanzi et al. (2011) introduced vulnerability of computation of square roots modulo the Goppa polynomial in the decryption process. For the computation, the suggestion of Huber (1996) is used. The leakage is presented in a polynomial multiplication in the algorithm of Huber (1996). By misusing this vulnerability, one can reveal the Goppa polynomial. Finally, they discuss next possibility how to reveal the Goppa polynomial, namely several leakage pints of the Extended Euclidean Algorithm used to solve the key equation in the Patterson's algebraic decoding algorithm.

Structural Attacks

Structural attacks, as one can deduce from the expression, are focused on the specific structure of error-correcting codes. The idea of using some codes with specific structure comes from the key size issue. There have been many proposals attempting to reduce the key size by altering the underlying error-correcting code. Often, the authors used highly structured codes which can be stored more efficiently (Section Code-based cryptography survey). And for detailed survey of the structural attacks before year 2011, consult Cayrel et al. (2011). As can be observed from the mentioned sources, and from the recent works, only the binary irreducible Goppa codes with maximal length (Section Post-quantum cryptography based on error-correcting codes) remain untouched. From the recent works, we note out that Landais, and Tillich (2013) proposed a structural attack against cryptosystems using convolutional codes. Next new structural attack is proposed by Gaborit, Ruatta, and Schrek (2013); it is aimed against cryptosystems based on rank metric codes, and thus aimed against cryptosystems based on generic rank syndrome decoding problem.

Information Set Decoding Attacks

Again, we refer a reader to the survey of Cayrel et al. (2011) for seeing what had been done before year 2011. From the very recent works we point out that Hamdaoui, and Sendrier (2013) proposed a non-asymptotic analysis of the Stern-Dumer variant, the May-Meurer-Thomae variant (May,

Meurer, & Thomae, 2011), and the Becker-Joux-May-Meurer variant (Becker et al., 2012) of the generic information set decoding attack. They stated that if the error weight is less or approximately equal to the Gilbert-Varshamov distance, the best attack is information set decoding attack. But when it is vice versa then it cannot be predicted whether the information set decoding attack, or generalized birthday algorithm is better.

Distinguisher Based Attacks

Faugère et al. (2013) presented a Goppa code distinguisher. Provided that the code rate (code dimension divided by code length) is very high, the distinguisher algorithm allows distinguishing Goppa codes from random codes. Next, Gauthier, Otmani, and Tillich (2012), and Couvreur et al. (2013) showed that even the choice of Generalized Reed-Solomon codes proposed in (Baldi et al., 2011) is not secure, as we mentioned in Section Code-based cryptography survey.

Quantum Attacks

The fastest quantum cryptanalytic attack against code-based cryptography can be found in (Bernstein, 2010). It is a quantum structural attack. Its complexity is still at least sub-exponential in case of the post-quantum cryptosystems.

CONCLUSION

We saw that many contributions have been made in the past three years. Construction of cryptographic primitives has been improved. Unfortunately, the key size issue has not been resolved yet. Almost all the attempts to change the underlying error-correcting code (a cryptographic primitive is based on) have failed regarding the recent cryptanalysis. On the other hand, two prospective countermeasures that can hide the structure of a secret error-correcting code have been proposed. But the only credible post-quantum constructions are those using the binary irreducible Goppa codes as the principal building block. We showed that implementations of post-quantum error-correcting code-based cryptography (namely McEliece and Niederreiter PKC) can be even better performing than the ECC. But still, there are many opportunities for optimization and standardization. Although the post-quantum cryptography is very resistant to attacks even employing post-quantum cryptanalysis, from the view of side channel attacks it is still equally vulnerable.

REFERENCES

Avanzi, R., Hoerder, S., Page, D., & Tunstall, M. (2011). Side-channel attacks on the McEliece and Niederreiter public-key cryptosystems. *Journal of Cryptographic Engineering*, *1*(4), 271–281. doi:10.1007/s13389-011-0024-9

Baldi, M., Bianchi, M., Chiaraluce, F., Rosenthal, J., & Schipani, D. (2011). A variant of the McEliece cryptosystem with increased public key security. In *Proceedings of WCC 2011 - Workshop on coding and cryptography* (pp. 173-182). Paris, France: Inria.

Baldi, M., Bianchi, M., Chiaraluce, F., Rosenthal, J., & Schipani, D. (2011). Enhanced public key security for the McEliece cryptosystem. *CoRR*. Retrieved August 11, 2011, from http://arxiv.org/abs/1108.2462

Baldi, M., Bodrato, M., & Chiaraluce, F. (2008). A New Analysis of the McEliece Cryptosystem Based on QC-LDPC Codes. In *Proceedings of 6th International Conference, SCN 2008* (LNCS), (pp. 246-262). Amalfi, Italy: Springer.

Barreto, P. S. L. M., Misoczki, R., & Simplício, M. A. Jr. (2011). One-Time Signature Scheme from Syndrome Decoding over Generic Error-Correcting Codes. *Journal of Systems and Software*, *84*(2), 198–204. doi:10.1016/j.jss.2010.09.016

Becker, A., Joux, A., May, A., & Meurer, A. (2012). Decoding random binary linear codes in 2n/20: How 1+1=0 improves information set decoding. In *Proceedings of Advances in Cryptology - EUROCRYPT (LNCS)* (Vol. 7237, pp. 520–536). Cambridge, UK: Springer.

Berlekamp, E., McEliece, R., & van Tilborg, H. (1978). On the inherent intractability of certain coding problems. *IEEE Transactions on Information Theory*, *24*(3), 384–386. doi:10.1109/TIT.1978.1055873

Berlekamp, E. R. (1971). Factoring polynomials over large finite fields. In *Proceedings of the second ACM symposium on Symbolic and algebraic manipulation (SYMSAC '71)*. New York: ACM.

Bernstein, D. J. (2010). Grover vs. McEliece. In *Proceedings of Post-Quantum Cryptography, Third International Workshop, PQCrypto 2010,* (pp. 73-80). Darmstadt, Germany: Springer.

Bernstein, D. J. (2011a). List decoding for binary Goppa codes. In *Proceedings of Coding and cryptology-third international workshop, IWCC 2011,* (LNCS), (vol. 6639, pp. 62-80). Qingdao, China: Springer.

Bernstein, D. J. (2011b). Simplified high-speed high-distance list decoding for alternant codes. In *Proceedings of the 4th international conference on Post-Quantum Cryptography. PQCrypto'11* (pp. 200-216). Taipei, Taiwan: Springer.

Bernstein, D. J., Buchmann, J., & Dahmen, E. (Eds.). (2008). *Post-Quantum Cryptography*. Springer.

Bernstein, D. J., Lange, T., & Peters, C. (2008). Attacking and Defending the McEliece Cryptosystem. In *Proceedings of the 2nd International Workshop on Post-Quantum Cryptography PQCrypto '08* (pp. 31-46). Cincinnati, OH: Springer Berlin Heidelberg.

Bernstein, D. J., Lange, T., & Peters, C. (2010). Wild McEliece. *Cryptology ePrint Archive: Report 2010/410*. Retrieved Jul 22, 2010, from http://eprint.iacr.org/2010/410

Bernstein, D. J., Lange, T., & Peters, C. (2011a). Wild McEliece Incognito. In *Proceedings of Post-Quantum Cryptography - 4th International Workshop, PQCrypto 2011* (pp. 244-254). Taipei, Taiwan: Springer.

Bernstein, D. J., Lange, T., & Peters, C. (2011b). Smaller decoding exponents: ball-collision decoding. In *Proceedings of 31st Annual Cryptology Conference,* (pp. 743-760). Santa Barbara, CA: Springer.

Bernstein, D. J., Lange, T., Peters, C., & Schwabe, P. (2011). Faster 2-regular Information-Set Decoding. In *Proceedings of the Third international conference on Coding and cryptology (IWCC'11)* (pp. 81-98). Qingdao, China: Springer.

Bernstein, D. J., Lange, T., Peters, C., & Schwabe, P. (2011). Really Fast Syndrome-Based Hashing. *Cryptology ePrint Archive: Report 2011/074*. Retrieved February 14, 2011, from http://eprint.iacr.org/2011/074

Beuchat, J.-L., Sendrier, N., Tisserand, A., & Villard, G. (2004). FPGA Implementation of a Recently Published Signature Scheme. *Research Report INRIA*. Retrieved 2004, from http://hal.inria.fr/docs/00/07/70/45/PDF/RR-5158.pdf

Biasi, F. P., Barreto, P. S. L. M., Misoczki, R., & Ruggiero, W. V. (2012). Scaling efficient code-based for embedded platforms. *CoRR*. Retrieved December 18, 2012, from http://arxiv.org/abs/1212.4317

Biswas, B., & Sendrier, N. (2008). McEliece Cryptosystem Implementation: Theory and Practice. In *Proceedings of Post-Quantum Cryptography Second International Workshop, PQCrypto 2008* (pp. 47-62). Cincinnati, OH: Springer.

Carlos, A., Gaborit, P., & Schrek, J. (2011). A new zero-knowledge code based identification scheme with reduced communication. *CoRR*. Retrieved November 7, 2011, from http://arxiv.org/abs/1111.1644

Cayrel, P.-L. (2012). Code-based cryptosystems: implementations. *cayrel.net: Code based cryptography*. Retrieved 2012, from http://cayrel.net/research/code-based-cryptography/code-based-cryptosystems/

Cayrel, P.-L., Alaoui, S. M. E. Y., Hoffmann, G., & Véron, P. (2012). An improved threshold ring signature scheme based on error correcting codes. In *Proceedings of Arithmetic of Finite Fields - 4th International Workshop, WAIFI 2012,* (pp. 45-63). Bochum, Germany: Springer.

Cayrel, P.-L., & Dusart, P. (2010). McEliece/Niederreiter PKC: Sensitivity to Fault Injection. In *Proceedings of FEAS, 2010 5th International Conference on Future Information Technology* (pp. 1-6). Busan, Korea: IEEE.

Cayrel, P.-L., ElYousfi, M., Hoffmann, G., Meziani, M., & Niebuhr, R. (2011). Recent progress in code-based cryptography. In *Proceedings of Information Security and Assurance - International Conference,* (pp. 21-32). Brno, Czech Republic: Springer.

Cayrel, P.-L., Gaborit, P., & Prouff, E. (2008). Secure Implementation of the Stern Authentication and Signature Schemes for Low-Resource Devices. In *Proceedings of 8th IFIP WG 8.8/11.2 International Conference, CARDIS 2008,* (pp. 191-205). London, UK: Springer.

Cayrel, P.-L., Hoffmann, G., & Persichetti, E. (2012). Efficient Implementation of a CCA2-Secure Variant of McEliece Using Generalized Srivastava Codes. In *Proceedings of 15th International Conference on Practice and Theory in Public Key Cryptography,* (pp. 138-155). Darmstadt, Germany: Springer.

Cayrel, P.-L., Véron, P., & Alaoui, S. M. E. Y. (2011). A zero-knowledge identification scheme based on the q-ary syndrome decoding problem. In *Proceedings of the 17th international conference on Selected areas in cryptography SAC'10* (171-186). Waterloo, Canada: Springer.

Chien, R. (1964). Cyclic decoding procedures for Bose-Chaudhuri-Hocquenghem codes. *IEEE Transactions on Information Theory*, *10*(4), 357–363. doi:10.1109/TIT.1964.1053699

Chizhov, I. V., & Borodin, M. A. (2013). The failure of McEliece PKC based on Reed-Muller codes. *Cryptology ePrint Archive: Report 2013/287*. Retrieved May 15, 2013, from http://eprint.iacr.org/2013/287

Couvreur, A., Gaborit, P., Gauthier, V., Otmani, A., & Tillich, J. P. (2013). Distinguisher-based attacks on public-key cryptosystems using Reed-Solomon codes. *CoRR*. Retrieved Jul 24, 2013, from http://arxiv.org/abs/1307.6458

Dottling, N., Dowsley, R., Muller-Quade, J., & Nascimento, A. C. A. (2012). A CCA2 Secure Variant of the McEliece Cryptosystem. *IEEE Transactions on Information Theory*, *58*(10), 6672–6680. doi:10.1109/TIT.2012.2203582

Dowsley, R., Müller-Quade, J., & Nascimento, A. C. A. (2009). A CCA2 Secure Public Key Encryption Scheme Based on the McEliece Assumptions in the Standard Model. *Lecture Notes in Computer Science*, *5473*, 240–251. doi:10.1007/978-3-642-00862-7_16

Eisenbarth, T., Güneysu, T., Heyse, S., & Paar, C. (2009). MicroEliece: McEliece for Embedded Devices. In *Proceedings of Cryptographic Hardware and Embedded Systems - CHES 2009, 11th International Workshop,* (pp. 49-64). Lausanne, Switzerland: Springer.

Faugère, J.-C., Gauthier-Umaña, V., Otmani, A., Perret, L., & Tillich, J.-P. (2013). A Distinguisher for High Rate McEliece Cryptosystems. *IEEE Transactions on Information Theory*, *59*(10), 6830–6844. doi:10.1109/TIT.2013.2272036

Faugère, J.-C., Otmani, A., Perret, L., & Tillich, J.-P. (2010). Algebraic Cryptanalysis of McEliece Variants with Compact Keys. In *Proceedings of the 29th Annual international conference on Theory and Applications of Cryptographic Techniques EUROCRYPT'10* (pp. 279-298). Springer.

Faure, C., & Minder, L. (2008). Cryptanalysis of the McEliece cryptosystem over hyperelliptic curves. In *Proceedings of the eleventh International Workshop on Algebraic and Combinatorial Coding Theory* (pp. 99-107). Pamporovo, Bulgaria: ACCT.

Finiasz, M. (2010). Parallel-CFS: Strengthening the CFS Mc-Eliece-Based Signature Scheme. In *Proceedings of Selected Areas in Cryptography - 17th International Workshop,* (pp. 159-170). Waterloo, Canada: Springer Berlin Heidelberg.

Finiasz, M., & Sendrier, N. (2009). Security Bounds for the Design of Code-based Cryptosystems. In *Proceedings of Advances in Cryptology - ASIACRYPT 2009, 15th International Conference on the Theory and Application of Cryptology and Information Security,* (pp. 88-105). Tokyo, Japan: Springer.

Fischer, J.-B., & Stern, J. (1996). An efficient pseudo-random generator provably as secure as syndrome decoding. In *Proceedings of the 15th annual international conference on Theory and application of cryptographic techniques EUROCRYPT'96* (pp. 245-255). Saragossa, Spain: Springer.

Gaborit, P., Ruatta, O., & Schrek, J. (2013). On the complexity of the Rank Syndrome Decoding problem. *CoRR*. Retrieved January 6, 2013, from http://arxiv.org/abs/1301.1026

Gaborit, P., & Schrek, J. (2012). Efficient code-based one-time signature from automorphism groups with syndrome compatibility. In *Proceedings of IEEE International Symposium on Information Theory, ISIT 2012* (pp. 1982-1986). Cambridge, MA: IEEE.

Gauthier, V., & Leander, G. (2009). Practical Key Recovery Attacks on Two McEliece Variants. *Cryptology ePrint Archive: Report 2009/509*. Retrieved October 21, 2009, from eprint.iacr.org/2009/509.pdf

Gauthier, V., Otmani, A., & Tillich, J.-P. (2012). A distinguisher-based attack on a variant of McEliece's cryptosystem based on Reed-Solomon codes. *CoRR*. Retrieved April 29, 2012, from http://arxiv.org/abs/1204.6459

Gauthier, V., Otmani, A., & Tillich, J.-P. (2012). A Distinguisher-Based Attack of a Homomorphic Encryption Scheme Relying on Reed-Solomon Codes. *Cryptology ePrint Archive: Report 2012/168*. Retrieved March 29, 2012, from http://eprint.iacr.org/2012/168

Gilbert, E. N. (1952). A comparison of signaling alphabets. *The Bell System Technical Journal*, *31*(3), 504–522. doi:10.1002/j.1538-7305.1952.tb01393.x

Goppa, V. D. (1970). A new class of linear error-correcting codes. *Probl. Peredach. Inform.*, *6*(3), 24–30.

Gueye, C. T., & Mboup, E. H. M. (2013). Secure Cryptographic Scheme based on Modified Reed Muller Codes. *International Journal of Security and Its Applications*, *7*(3), 5.

Helgert, H. J. (1974). Alternant Codes. *Information and Control*, *26*(4), 369–380. doi:10.1016/S0019-9958(74)80005-7

Heyse, S. (2010). Low-Reiter: Niederreiter Encryption Scheme for Embedded Microcontrollers. In *Proceedings of Post-Quantum Cryptography, Third International Workshop, PQCrypto 2010,* (pp. 165-181). Darmstadt, Germany: Springer.

Heyse, S., & Güneysu, T. (2012). Towards One Cycle per Bit Asymmetric Encryption: Code-Based Cryptography on Reconfigurable Hardware. In *Proceedings of Cryptographic Hardware and Embedded Systems - CHES 2012 - 14th International Workshop,* (pp. 340-355). Leuven, Belgium: Springer.

Heyse, S., & Güneysu, T. (2013). Code-based cryptography on reconfigurable hardware: tweaking Niederreiter encryption for performance. *Journal of Cryptographic Engineering*, *3*(1), 29–43. doi:10.1007/s13389-013-0056-4

Heyse, S., Moradi, A., & Paar, C. (2010). Practical Power Analysis Attacks on Software Implementations of McEliece. In *Proceedings of Post-Quantum Cryptography, Third International Workshop, PQCrypto 2010,* (pp. 108-125). Darmstadt, Germany: Springer.

Heyse, S., von Maurich, I., & Güneysu, T. (2013). Smaller Keys for Code-Based Cryptography: QC-MDPC McEliece Implementations on Embedded Devices. In *Proceedings of Cryptographic Hardware and Embedded Systems - CHES 2013 - 15th International Workshop,* (pp. 273-292). Santa Barbara, CA: Springer.

Huber, K. (1996). Note on decoding binary Goppa codes. *Electronics Letters*, *32*(2), 102–103. doi:10.1049/el:19960072

Kobara, K., & Imai, H. (2001). Semantically Secure McEliece Public-Key Cryptosystems-Conversions for McEliece PKC. In *Proceedings of the 4th International Workshop on Practice and Theory in Public Key Cryptography: Public Key Cryptography (PKC '01).* London, UK: Springer.

Landais, G., & Tillich, J.-P. (2013). An Efficient Attack of a McEliece Cryptosystem Variant Based on Convolutional Codes. In *Proceedings of 5th International Workshop, PQCrypto 2013,* (pp. 102-117). Limoges, France: Springer.

Lee, P. J., & Brickell, E. F. (1988). An observation on the security of McEliece's public-key cryptosystem. *Lecture Notes in Computer Science*, 275–280. doi:10.1007/3-540-45961-8_25

Lu, R., Lin, X., Liang, X., & Shen, X. S. (2011). An efficient and provably secure public-key encryption scheme based on coding theory. *Security and Communication Networks*, *4*(12), 1440–1447. doi:10.1002/sec.274

Mathew, K. P., Vasant, S., & Rangan, C. P. (2012). On Provably Secure Code-based Signature and Signcryption Scheme. *Cryptology ePrint Archive, Report 2012/585*. Retrieved October 15, 2012, from http://eprint.iacr.org/2012/585

Mathew, K. P., Vasant, S., Venkatesan, S., & Rangan, C. (2012). An Efficient IND-CCA2 Secure Variant of the Niederreiter Encryption Scheme in the Standard Model. In *Proceedings of 17th Australasian Conference, ACISP 2012,* (pp. 166-179). Wollongong, Australia: Springer.

May, A., Meurer, A., & Thomae, E. (2011). Decoding random linear codes in O(2^0.054n). In *Proceedings of Advances in Cryptology - ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security,* (pp. 107–124). Seoul, South Korea: Springer.

McEliece, R. (1978). A Public-Key Cryptosystem Based on Algebraic Coding Theory. *Deep Space Network Progress Report, DSN PR 42–44,* NASA Code 310-10-67-11. Retrieved April 15, 1978, from http://ipnpr.jpl.nasa.gov/progress_report2/42-44/44title.htm

Meziani, M., Hoffmann, G., & Cayrel, P.-L. (2012). Improving the Performance of the SYND Stream Cipher. In *Proceedings of Progress in Cryptology - AFRICACRYPT 2012 - 5th International Conference on Cryptology in Africa,* (pp. 99-116). Ifrane, Morocco: Springer.

Misoczki, R., Tillich, J.-P., Sendrier, N., & Barreto, P. S. L. M. (2013). MDPC-McEliece: New McEliece Variants from Moderate Density Parity-Check Codes. In *Proceedings of the 2013 IEEE International Symposium on Information Theory,* (pp. 2069-2073). Istanbul, Turkey: IEEE.

Molter, H. G., Stöttinger, M., Shoufan, A., & Strenzke, F. (2011). A simple power analysis attack on a McEliece cryptoprocessor. *Journal of Cryptographic Engineering*, *1*(1), 29–36. doi:10.1007/s13389-011-0001-3

Niederreiter, H. (1986). Knapsack-type Cryptosystems and Algebraic Coding Theory. *Problems of Control and Information Theory*, *15*(2), 159–166.

Overbeck, R. (2008). Structural Attacks for Public Key Cryptosystems Based on Gabidulin Codes. *J. Cryptology*, *21*(2), 280–301. doi:10.1007/s00145-007-9003-9

Overbeck, R. (2008). An Analysis of Side Channels in the McEliece PKC. *Enhancing Crypto-Primitives with Techniques from Coding Theory. NATO OTAN*. Retrieved 2008, from https://www.cosic.esat.kuleuven.be/nato arw/slides participants/Overbeck slides nato08.pdf

Overbeck, R., & Sendrier, N. (2008). Code-Based Cryptography. In *Post-Quantum Cryptography* (pp. 95–145). Springer.

Patterson, N. (1975). The algebraic decoding of Goppa codes. *IEEE Transactions on Information Theory*, *21*(2), 203–207. doi:10.1109/TIT.1975.1055350

Persichetti, E. (2012). On a CCA2-secure variant of McEliece in the standard model. *IACR Cryptology ePrint Archive*. Retrieved May 11, 2012, from http://eprint.iacr.org/2012/268

Peters, C. (2010). Information-Set Decoding for Linear Codes over F_q. In *Proceedings of Post-Quantum Cryptography, Third International Workshop, PQCrypto 2010,* (pp. 81-94). Darmstadt, Germany: Springer.

Proos, J., & Zalka, C. (2003). Shor's discrete logarithm quantum algorithm for elliptic curves. *Quantum Info. Comput.*, *3*(4), 317–344.

Rastaghi, R. (2013). An Efficient CCA2-Secure Variant of the McEliece Cryptosystem n the Standard Model. *CoRR*. Retrieved February 2, 2013, from http://arxiv.org/abs/1302.0347

Sendrier, N. (2000). Finding the permutation between equivalent linear codes: the support splitting algorithm. *IEEE Transactions on Information Theory*, *46*(4), 1193–1203. doi:10.1109/18.850662

Shor, P. W. (1997). Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM Journal on Computing*, *26*(5), 1484–1509. doi:10.1137/S0097539795293172

Shoufan, A., Strenzke, F., Molter, H. G., & Stöttinger, M. (2009). A Timing Attack against Patterson Algorithm in the McEliece PKC. In *Proceedings of Information, Security and Cryptology - ICISC 2009, 12th International Conference,* (pp. 161-175). Seoul, Korea: Springer.

Shoufan, A., Wink, T., Molter, G., Huss, S., & Strenzke, F. A. (2009). Novel Processor Architecture for McEliece Cryptosystem and FPGA Platforms. In *Proceedings of Application-specific Systems, Architectures and Processors* (pp. 98–105). Boston, MA: IEEE. doi:10.1109/ASAP.2009.29

Sidelnikov, V. M., & Shestakov, S. (1992). On Cryptosystems based on Generalized Reed-Solomon Codes. *Discrete Mathematics*, *4*(3), 57–63.

Sidelnikov, V. M., & Shestakov, S. O. (1992). On insecurity of cryptosystems based on generalized Reed-Solomon codes. *Discrete Mathematics*, *2*(4), 439–444.

Stern, J. (1994a). A New Identification Scheme Based on Syndrome Decoding. In *Proceedings of the 13th Annual International Cryptology Conference on Advances in Cryptology* (pp. 13-21). Santa Barbara, CA: Springer.

Stern, J. (1994b). Designing Identification Schemes with Keys of Short Size. In *Proceedings of Advances in Cryptology – Proceedings of CRYPTO '94,* (vol. 839, pp. 164-173). Santa Barbara, CA: Springer Berlin Heidelberg.

Strenzke, F. (2010). A Smart Card Implementation of the McEliece PKC. In *Proceedings of 4th IFIP WG 11.2 International Workshop, WISTP 2010,* (pp. 47-59). Passau, Germany: Springer.

Strenzke, F. (2010). A Timing Attack against the Secret Permutation in the McEliece PKC. In *Proceedings of Third International Workshop, PQCrypto 2010,* (pp. 95-107). Darmstadt, Germany: Springer.

Strenzke, F. (2011). Fast and Secure Root-Finding for Code-based Cryptosystems. *IACR Cryptology ePrint Archive*. Retrieved December 11, 2011, from http://eprint.iacr.org/2011/672

Strenzke, F. (2012). Solutions for the Storage Problem of McEliece Public and Private Keys on Memory-Constrained Platforms. In *Proceedings of 15th International Conference, ISC 2012,* (pp. 120-135). Passau, Germany: Springer.

Strenzke, F. (2013). Timing Attacks against the Syndrome Inversion in Code-Based Cryptosystems. In *Proceedings of 5th International Workshop, PQCrypto 2013,* (pp. 217-230). Limoges, France: Springer.

Strenzke, F., Tews, E., Molter, H. G., Overbeck, R., & Shoufan, A. (2008). Side Channels in the McEliece PKC. In *Proceedings of the Second international Workshop on Post-Quantum Cryptography PQCRYPTO 2008,* (LNCS), (pp. 216-229). Cincinnati, OH: Springer.

Sugiyama, Y., Kasahara, M., Hirasawa, S., & Namekawa, T. (1976). An erasures-and-errors decoding algorithm for Goppa codes. *IEEE Transactions on Information Theory*, *22*(2), 238–241. doi:10.1109/TIT.1976.1055517

von Maurich, I., & Güneysu, T. (2012). Embedded Syndrome-Based Hashing. In *Proceedings of Progress in Cryptology - INDOCRYPT 2012, 13th International Conference on Cryptology in India,* (pp. 339-357). Kolkata, India: Springer.

Yang, B.-Y. (Ed.). (2011). *Proceedings of 4th International Workshop*. Taipei, Taiwan: Springer.

Zalka, C. (1998). Fast versions of Shor's quantum factoring algorithm. *Coronell University Library arXiv.org*. Retrieved 24 June 1998, from http://arxiv.org/abs/quant-ph/9806084

KEY TERMS AND DEFINITIONS

Code-Based Cryptography: Cryptography that is based on an NP-complete problem in coding theory, namely General decoding problem, or Syndrome decoding problem.

General Decoding Problem: Given a code generator matrix (or parity check matrix equivalently) and a vector that is not from the code, try to correct and decode the vector.

Identification: Or also Authentication is technique (real-time-) verifying identity of one entity asking for a secret that is able to answer correctly only the entity of identity that is claimed.

NP-Complete Problem: It is a decision problem that is as hard as any nondeterministic polynomial time problem, i.e. the decision running time for a NP-complete problem is at least sub-exponential.

Post-Quantum Cryptography: Cryptography for which the best know attack has at least a sub-exponential complexity even using quantum cryptanalysis. Post-quantum Cryptography can be Code-based, Hash-based, Lattice-based, or Multivariate-quadratic-equation based.

Public Key Cryptosystem: An asymmetric cryptosystem that uses a key pair, namely private key and corresponding public key.

Side Channel Attacks: Attacks that exploits information obtained from any source, not only cipher-text or plaintext.

Signature: Stands for digital signature process applied onto a message resulting in a data string that is afterwards used in order to verify integrity, authenticity, and signatory non-repudiation.

Syndrome Based Cryptography: Code-based cryptography based on syndrome decoding problem.

Syndrome Decoding Problem: Given a parity check matrix, and a syndrome find the corresponding error vector.