Flexible Quasi-Dyadic Code-Based Public-Key Encryption and Signature

Kazukuni Kobara

Research Center for Information Security (RCIS), National Institute of Advanced Industrial Science and Technology (AIST), 10F 1003, Akihabara-Daibiru, 1-18-13, Soto-Kanda, Tiyoda-ku, Tokyo, 101-0021, Japan.
`kobara_conf@m.aist.go.jp`

Abstract. Drawback of code-based public-key cryptosystems is that their public-key size is lage. It takes some hundreds KB to some MB for typical parameters. While several attempts have been conducted to reduce it, most of them have failed except one, which is Quasi-Dyadic (QD) public-key (for large extention degrees). While an attack has been proposed on QD public-key (for small extension degrees), it can be prevented by making the extension degree m larger, specifically by making $q^{(m(m-1))}$ large enough where q is the base filed and for a binary code, $q = 2$. The drawback of QD is, however, it must hold $n << 2^m - t$ (at least $n \leq 2^{m-1}$) where n and t are the code lenght and the error correction capability of the underlying code. If it is not satisfied, its key generation fails since it is performed by trial and error. This condition also prevents QD from generating parameters for code-based digital signatures since without making n close to $2^m - t$, $2^{mt}/\binom{n}{t}$ cannot be small. To overcome these problems, we propose "Flexible" Quasi-Dyadic (FQD) public-key that can even achieve $n = 2^m - t$ with one shot. Advantages of FQD include 1) it can reduce the public-key size further, 2) it can be applied to code-based digital signatures, too.

Keywords public-key, digital signature, linear code, dyadic

1 Introduction

Public-key cryptosystems (PKCs) can be divided into the categories[1] shown in Fig. 1 and 2, respectively. Almost all of the currently deployed ones are based only on a small class of hard problems, namely Integer Factoring Problem (IFP) or Discrete Logarithm Problem (DLP). They are referred to as number theoretic problems. The number theoretic problem based PKCs have the following disadvantages that should be solved in short term and long term, respectively. The long term problem is the lack of quantum tolerance. The number theoretic problems are closely related to a problem to determine the cycle (hence they may be referred to as a cyclic problem) and they will be solved in (probabilistic) polynomial-time after the emergence of quantum computers [24] though

[1] Multivariate polynomial based ones may be included, but all of them have been broken and no relief method is known so far.

Integer Factoring Based:
- RSA
- Rabin
- Okamoto-Uchiyama
- Paillier

Discrete Logarithm Based:
- Diffie-Hellman
- ElGamal
- ECC
- XTR
- Cramer-Shoup
- Kurosawa-Desmedt

Fig. 1. Examples of PKCs Based on Number Theoretic (Cyclic) Problem

Code Based:
- McEliece
- Niederreiter

Lattice Based:
- NTRU
- Ajtai-Dwork
- Goldreich-Goldwasser-Halevi
- Ajtai
- Regev
- Peikert

Subset Sum Based:
- Okamoto-Tanaka-Uchiyama

Fig. 2. Examples of PKCs Based on Combinatorial Problem

several breakthroughs are needed to realize quantum computers. The short term problem is the requirement of heavy multiple precision modular exponentiations that are not easy to deploy with low cost on low-computational power devices, such as RFID (Radio Frequency Identity), sensors and SCADA (Supervisory Control And Data Acquisition) devices.

On the other hand, combinatorial-problems are quantum tolerant and only small arithmetic units, e.g. addition in a small field or ring, are required for encryption and signature verification. Furthermore, among the combinatorial-problem based PKCs, code-based PKCs are advantageous in redundancy, i.e. (Plaintext Size) – (Ciphertext Size), and in the arithmetic unit, i.e. encryption and signature verification consists mostly on exclusive-ors that are highly parallelizable. Hence, code-based PKCs are suitable for heterogeneous applications where one side may have a reasonable computational power, but that of the other side is limited such as privacy-preserving RFID [8] and lightweight broadcast authentication for emergency. Other than them, code-based primitives can be utelized to construct ZKIP (Zero Knowledge Interactive Proof) [26], hash functions [2], OT (Oblivious Transfer) [17, 11] and so on.

The strongest security notion of a PKC, IND-CCA2 (Indistinguishability against Adaptive Chosen Ciphertext Attack), can be achieved by applying “appropriate” conversion scheme to the primitive code-based PKEs as long as it satisfies OW-CPA (One-Wayness against Chosen Plaintext Attack). For the McEliece primitive PKC, specific conversion scheme [15] makes the redundancy smallest while maintaining provable security in the random oracle model. For the Niederreiter primitive PKC, either OAEP++ [14] for a long plaintext or OAEP+ [25] for a small plaintext can achieve them. Not only in the random oracle model, provable security of IND-CPA and IND-CCA2 have been achieved in the standard model in [23] and [10] respectively even though the constructions in the standard model are less efficient compared to those in the random oracle model. Anyway, secure constructions are available as long as the underlying primitive code-based PKCs satisfy OW-CPA and the parameters meeting OW-CPA are estimated in [12] against the most powerful attacks (Optimized) Information Set Decoding (OISD[2]) and Generalized Birthday Attack (GBA).

The drawback of code-based PKCs is, however, that the publick-key size is large, which is $k(n-k)$ bits if a binary code of length n with information rate k/n is used. To overcome this problem, several attempts have been conducted. They are summarized as follows.

(Potential approaches for reducing public-key size for code-based PKCs)
Enhancement of error correction capability:

- Capacity Approaching Codes
  - LDPC codes
  - QC-LDPC codes [3]
- List Decoding
  - Exhaustive search
  - List decoding for Goppa Code [6]
- Error expansion/hold [18]

Compression of public-key:

- Quasi-Cyclic Construction [4]
- Quasi-Dyadic Construction [21]
- Flexible-Quasi-Dyadic Construction (proposal)

Unfortunately, LDPC (Low-Density Parity Check) code approach has been broken in [22, 13] where [22] works if the density of the random nonsingular secret matrix S is low and [13] works for any S. Error expansion/hold approach has been broken in [16]. Quasi-Cyclic and QC-LDPC approaches have been broken in [1, 28]. Quasi-Dyadic approach has been broken in [28], but only for small extension degrees [20]. Hence the remaining approaches are list decoding and Quasi-Dyadic approach for large extension degrees. While list decoding works, its effect is small since it can correct only a couple of more errors for practical parameters within practical decoding complexity. Hence the last resort is the quasi-dyadic approach with large extension degrees.

2 Quasi-Dyadic Construction

I will skip the preliminary of code-based PKCs, but you can find a lot of contents to explain them, e.g. in the surveys section of [5] or in [9].

Quasi-Dyadic construction was proposed in [21]. It uses the inter section between dyadic matrices and Goppa codes in Cauchy form. A $2^\nu \times 2^\nu$ dyadic matrix M is in this form:

$$M = \begin{bmatrix} A\ B \\ B\ A \end{bmatrix} \tag{1}$$

where A and B are $2^{\nu-1} \times 2^{\nu-1}$ dyadic matrices, respectively. The advantage of a dyadic matrix is that the whole matrix can be constructed from its one row or one column. This is the trick to reduce the public matrix.

[2] In [12], it is referred to as ISD but in this paper we call it OISD to distinguish it from classical ISDs.

Table 1. Sample parameters of plain code-based PKE estimated in [12]

m	t	n	BWF OISD $(p.l)$	Public-key size	Plaintext/Ciphertext
11	32	$2,048$	$2^{86.8}$ $(4,24)$	72.9KB	$233/352$ [bits]
12	41	$4,098$	$2^{128.5}$ (10.54)	216.5KB	$327/492$ [bits]

Table 2. Sample parameters of Quasi-Dyadic (QD) code-based PKE [21]

m	t	n	BWF OISD $(p.l)$	Public-key size	Plaintext/Ciphertext
16	64	$2,560$	$2^{91.3}$ $(1,12)$	3.0KB	$427/1024$ [bits]
16	64	$3,072$	$2^{108.0}$ $(2,17)$	4.0KB	$445/1024$ [bits]
16	128	$4,096$	$2^{135.8}$ $(2,18)$	4.0KB	$817/1024$ [bits]

Due to the following Theorem, it is possible to make a parity check matrix of the Goppa code Cauchy from.

Theorem 1 (Goppa Codes in Cauchy Form [27, 19]) *The Goppa code generated by a monic polynomial $g(x) = (x - z_0)\cdots(x - z_{t-1})$ without multiple zeros admits a parity-check matrix H whose i-th row and j-th column is $H_{ij} = 1/(z_i - L_j)$ for $0 \le i < t$ and $0 \le j < n$.*

The Cauchy matrix can be dyadic by choosing distinct z_i and L_j meeting the following conditions:

$$\frac{1}{h_{i\oplus j}} = \frac{1}{h_i} + \frac{1}{h_j} + \frac{1}{h_0} \tag{2}$$

$$z_i = \frac{1}{h_i} + \omega \tag{3}$$

$$L_j = \frac{1}{h_j} + \frac{1}{h_0} + \omega \tag{4}$$

The construction algorithm proposed in [21] generates a sequence of h_i for $0 \le i \le N$ where $n < N$ at random meeting (2) to (4). If they are not satisfied, it discards h_i and regenerates them until the conditions are satisfied. Using the generated h_i, a $N \times N$ full dyadic matrix can be constructed. It finally picks up a $t \times n$ sub-matrix from the full $N \times N$ dyadic matrix.

This algorithm is, however, restrictive on its parameter choice, i.e. $n << 2^m - t$ must hold otherwise it eventually fails to generate a distinct set of z_i and L_j, or takes a lot of time since it generates them by trial-and-error. This restriction prevents it from generating parameters for digital signatures since in digital signatures $2^{mt}/\binom{n}{t}$ must be small enough and without making n close to $2^m - t$, $2^{mt}/\binom{n}{t}$ cannot be small.

3 Flexible-Quasi-Dyadic Construction

To overcome the problems in QD, we propose a more flexible and efficient construction, which we call Flexible-Quasi-Dyadic (FQD) construction. FQD does not use trial-and-

Table 3. Sample parameters of Flexible-Quasi-Dyadic (FQD) code-based PKE (proposal)

m	t	n	BWF OISD $(p.l)$	BWF UL	Public-key size	Plaintext/Ciphertext
11	32	$2,016$	$2^{86.0}$ $(4,24)$	-	2.2KB	$224/352$ [bits]
11	37	$1,984$	$2^{90.3}$ $(4,24)$	-	2.1KB	$262/407$ [bits]
11	64	$1,984$	$2^{103.1}$ $(4,25)$	-	1.7KB	$404/704$ [bits]
11	96	$1,920$	$2^{91.0}$ $(2,16)$	-	1.2KB	$546/1056$ [bits]
11	112	$1,920$	$2^{80.0}$ $(2,16)$	-	0.92KB	$546/1056$ [bits]
12	19	$4,064$	$2^{81.0}$ $(8,44)$	-	5.6KB	$171/228$ [bits]
12	23	$4,064$	$2^{91.4}$ $(8,44)$	-	5.5KB	$202/276$ [bits]
12	32	$4,064$	$2^{111.6}$ $(10,53)$	-	5.4KB	$266/384$ [bits]
12	42	$4,032$	$2^{129.3}$ $(9,49)$	-	5.2KB	$333/504$ [bits]
12	64	$4,032$	-	$2^{157.4}$	4.8KB	$470/768$ [bits]
12	128	$3,968$	-	$2^{156.4}$	3.6KB	$811/1536$ [bits]
12	186	$3,840$	-	$2^{155.9}$	2.4KB	$1069/2232$ [bits]
12	256	$3,840$	$2^{91.3}$ $(1,13)$	-	1.1KB	$1352/3072$ [bits]
12	256	$3,728$	$2^{80.0}$ $(1,13)$	-	0.96KB	$1340/3072$ [bits]

Table 4. Sample parameters of plain code-based signature (CFS signature [7])

m	t	n	BWF GBA	BWF OISD $(p.l)$	Public-key size	Iteration	Signature Size
19	11	$524,288$	$2^{83.6}$	-	$13,370.7$KB	$2^{25.3}$	209 (234.3) [bits]
15	12	$32,768$	$2^{81.5}$	-	716.0KB	$2^{28.8}$	180 (208.8) [bits]
15	13	$32,768$	$2^{84.8}$	-	775.4KB	$2^{32.5}$	195 (227.5) [bits]
14	14	$16,384$	-	$2^{84.0}$ $(11,66)$	387.3KB	$2^{36.4}$	196 (232.4) [bits]
14	15	$16,384$	-	$2^{89.2}$ $(11,67)$	414.6KB	$2^{40.3}$	210 (250.3) [bits]
13	16	$8,192$	-	$2^{83.5}$ $(9,52)$	202.7KB	$2^{44.3}$	208 (252.3) [bits]

error approach and generates distinct z_i and L_j with one shot even for $n = 2^m - t$. FQD does not have any restriction such as $n << 2^m - t$.

FQD construction is as follows. It firstly generates one small $u \times u$ dyadic matrix using δ_i for $0 \le i < \log_2 u$. We call them "inner delta" since they define the inner structure of the $u \times u$ full dyadic matrix. Then FQD generates the other $u \times u$ full dyadic matrices by duplicating the inner structure of the first $u \times u$ full dyadic matrix but shifting them using both Δ_{j_1} and Δ'_{i_1} for $0 \le j_1 < \lceil n/u \rceil$ and $1 \le i_1 < \lceil t/u \rceil$, respectively. We call Δ_{j_1} and Δ'_{i_1} "outer delta" since they define the relationship among the full $u \times u$ dyadic matrices. FQD can also remove the block-wise permutation and removal in the key generation phase of QD since the choice of Δ_{j_1} and n already includes them. This is another advantage of FQD.

Table 5. Sample parameters of Flexible-Quasi-Dyadic (FQD) code-based digital signature (proposal)

m	t	n	BWF GBA	BWF OISD $(p.l)$	Public-key size	Iteration	Signature Size
19	11	$524,272$	$2^{83.6}$	-	$1,215.5$KB	$2^{25.3}$	209 (234.3) [bits]
15	12	$32,752$	$2^{81.5}$	-	59.6KB	$2^{28.8}$	180 (208.8) [bits]
15	13	$32,752$	$2^{84.8}$	-	59.6KB	$2^{32.5}$	195 (227.5) [bits]
14	14	$16,368$	$2^{84.1}$	-	27.6KB	$2^{36.4}$	196 (232.4) [bits]
14	15	$16,368$	-	$2^{89.2}$ $(11,67)$	27.6KB	$2^{40.3}$	210 (250.3) [bits]
13	16	$8,176$	-	$2^{83.4}$ $(9,52)$	12.6KB	$2^{44.3}$	208 (252.3) [bits]

I will explain how to choose δ_i, Δ_{j_1} and Δ'_{i_1} later on, but once they are determined, z_i and L_j are given as follows:

$$z_{i_0} = \oplus_{b=0}^{\log_2 u-1} i_0[b] \cdot \delta_b \qquad \text{for} \quad 0 \le i_0 < u \tag{5}$$

$$z_{i_1 \cdot u+i_0} = z_{i_0} \oplus \Delta'_{i_1} \qquad \text{for} \quad 1 \le i_1 < \lceil t/u \rceil \tag{6}$$

$$L_{j_1 \cdot u+j_0} = z_{j_0} \oplus \Delta_{j_1} \qquad \text{for} \quad 0 \le j_1 < \lceil n/u \rceil \tag{7}$$

where $\oplus$ denotes exclusive-or, $i[b]$ and $j[b]$ denote $(b+1)$-th bit of i and j in the binary form, respectively. One can easily verify that $h_{i,j} = 1/(z_i \oplus L_j)$ makes a quasi-dyadic matrix. When $t \le u$, $z_{i_1 \cdot u+i_0}$ can be ignored. When $\lceil t/u \rceil \cdot u > t$ and/or $\lceil n/u \rceil \cdot u > n$, by removing $\lceil t/u \rceil \cdot u - t$ rows and $\lceil n/u \rceil \cdot u - n$ columns respectively, the size can be $t \times n$. Another option is to add removed z_i as L_j. This is useful to achieve $n = 2^m - t$ when $t \ne 2^x$ for any positive integer x.

The variables δ_i, Δ_{j_1} and Δ'_{i_1} must be chosen at random while making all the z_i for $0 \le i < t$ and L_j for $0 \le j < n$ distinct, i.e.

$$z_i \oplus z_{i'} \ne 0 \qquad \text{for } i \ne i' \tag{8}$$

$$L_j \oplus L_{j'} \ne 0 \qquad \text{for } j \ne j' \tag{9}$$

$$z_i \oplus L_j \ne 0 \tag{10}$$

These conditions are equivalent to the following conditions:

1. δ_b for $0 \le b < \log_2 u$ are linearly independent.
2. $\forall r \in \{0,1\}^{\log_2 u}$,

$$\Delta'_{i_1}, \Delta_{j_1}, (\Delta'_{i_1} \oplus \Delta_{j_1}), (\Delta'_{i_1} \oplus \Delta'_{i'_1}), (\Delta_{j_1} \oplus \Delta_{j'_1}) \notin \oplus_{b=0}^{\log_2 u-1} r[b] \cdot \delta_b \tag{11}$$

where $r[b]$ denotes the $(b+1)$-th bit of r in the binary form.

δ_b, Δ'_{i_1} and Δ_{j_1} satisfying the above conditions can be generated by the following algorithm:

1. Generate a $m \times m$ random binary nonsingular matrix M.
2. Let the $(b+1)$-th row from the top of M denote δ_b for $0 \le b \le (\log_2 u) - 1$.
3. Choose distinct Δ'_{i_1} and Δ_{j_1} from a linear combination of the bottom $m - \log_2 u$ rows of M.

The cardinality of a nonsingular matrix M is around $0.289 \cdot 2^{m^2}$, which is one of the secrets of FQD construction. Other secrets include permutation among Δ_{j_1}, random scalar multiplication with each $u \times u$ full dyadic block and multiplication of non-singular random dyadic matrix S.

We show some sample parameters for binary codes in Table 1 to 5, but the idea of FQD construction can easily be extended to non-binary codes, too. In these tables, m, t and n are parameters of the underlying code. m is the extension degree, t is the error correction capability and n is the code length. In plain (non-quasi-dyadic) schemes, $n = 2^m$ or $n < 2^m$, in QD, $n << 2^m - t$ and in FQD, $n = 2^m - t$ (or $n < 2^m - t$). BWF is the minimal binary workfactor to break the system, which is either Optimized Information Set Decoding (OISD), Generalized Birthday Attack (GBA) or the attack in [28] on QD/FQD (we call it UL attack). The values of OISD and GBA follow the estimation in [12]. p and l are optimum parameters for OISD. In [28], the BWF of UL, $\mathrm{BWF}_{\mathrm{UL}}$ is estimated as $q^2 \times (\log_2 q^2)^3 (v^2 + 3v + b)^2 v(v + b)$ where $v = \log_2 u$ and $b = \lceil n/u \rceil$, but this estimation is for $m = 2$. For $m \geq 2$, it is

$$\mathrm{BWF}_{\mathrm{UL}} = q^{m(m-1)} \times (\log_2 q^2)^3 (v^2 + 3v + b)^2 v(v + b) \tag{12}$$

In the columns of BWF "-" means the corresponding attack is less powerful. In the column of public-key size, KB$= 1024 \times 8$ bits. Plaintext/Ciphertext is the plaintext size and the ciphertext size in bits in the Niederreiter form. Iteration shows the signature generation cost, i.e. the number of trials to decode an error pattern corresponding to given syndromes. The signature size in () is when the error pattern is expressed as the positions of t errors. This increases the signature size but decreases the signature verification cost compared with the case where an error pattern is expressed as an integer between 0 and $\binom{n}{t} - 1$. The signature size can be reduced further by using the same technique in [7], i.e. by removing some error positions in the signature even though this increases the verification cost.

4 Conclusion

We proposed Flexible Quasi-Dyadic (FQD) construction, which can make the Quasi-Dyadic (QD) construction more flexible. FQD can achieve the maximum code length $n = 2^m - t$ with one shot whereas QD must hold $n << 2^m - t$ and its key generation is performed by trial and error. FQD's ability to make n close to $2^m - t$ is crucial for code-based digital signatures since without this ability $2^{mt} / \binom{n}{t}$ cannot be small and code-based digital signatures cannot be constructed. FQD can make n close to $2^m - t$ and can even be used to reduce the signature-verification-key size of code-based digital signatures.

Acknowledgment

The author would like to thank Paulo S. L. M. Barreto, Rafael Misoczki and Yang Cui for fruitful discussions on the attacks on Quasi-Dyadic public-keys.

References

1. L. Dallot A. Otmani, J.P. Tillich. “Cryptanalysis of two McEliece cryptosystems based on quasi-cyclic codes”. `http://arxiv.org/abs/0804.0409`, 2008.
2. D. Augot, M. Finiasz, Ph. Gaborit, S. Manuel, and N. Sendrier. “SHA-3 proposal: FSB”. SHA-3 NIST competition, 2008.
3. M. Baldi and F. Chiaraluce. “Cryptanalysis of a new instance of McEliece cryptosystem based on QC-LDPC codes”. In *Proc. of IEEE International Symposium on Information Theory, ISIT ’07*, pages 2591–2595, 2007.
4. T. Berger, P.-L. Cayrel, P. Gaborit, and A. Otmani. “Reducing key length of the McEliece cryptosystem”. In *Proc. of AFRICACRYPT ’09, LNCS 5580*, pages 77–79. Springer–Verlag, 2009.
5. D. J. Bernstein. “Code-based public-key cryptography”. `http://pqcrypto.org/code.html`.
6. D. J. Bernstein. “List decoding for binary Goppa codes”. `http://cr.yp.to/codes/goppalist-20081107.pdf`, 2008.
7. N. T. Courtois, M. Finiasz, and N. Sendrier. “How to achieve a McEliece-based digital signature scheme”. In *Proc. of ASIACRYPT 2001*, pages 157–174. Springer–Verlag, 2001.
8. Y. Cui, K. Kobara, K. Matsuura, and H. Imai. “Lightweight privacy-preserving authentication protocols secure against active attack in an asymmetric way”. *IEICE Trans.*, E91-D(5):1457–1465, 2008.
9. A. Schmidt D. Engelbert, R. Overbeck. “A summary of McEliece-type cryptosystems and their security”. *Journal of Mathematical Cryptology, (Previous version is available at `http://eprint.iacr.org/2006/162`)*, 1, 2007.
10. R. Dowsley, J. Muller-Quade, and A. C. A. Nascimento. “A CCA2 secure public key encryption scheme based on the McEliece assumptions in the standard model”. `http://eprint.iacr.org/2008/468`, 2008.
11. R. Dowsley, J. van de Graaf, J. M.-Quade, and A. Nascimento. “Oblivious transfer based on the McEliece assumptions”. In *Proc. of ICITS’08, LNCS 5155*, pages 107–117. Springer–Verlag, 2008.
12. M. Finiasz and N. Sendrier. “Security bounds for the design of code-based cryptosystems”. In *Proc. of ASIACRYPT 2009*, pages 88–105. Springer–Verlag, 2009.
13. M. Hagiwara, K. Kobara, and H. Imai. “On the security of McEliece public key cryptosystem with LDPC code (in japanese)”. In *The 2007 Symposium on Cryptography and Information Security : 2C1-1*, January 2007.
14. K. Kobara and H. Imai. “OAEP++ – another very simple way to fix the bug in OAEP –”. In *Proc. of 2002 International Symposium on Information Theory and Its Applications: S6-4-5*, pages 563–566, 2002.
15. K. Kobara and H. Imai. “Semantically secure McEliece public-key cryptosystem”. *IEICE Trans.*, E85-A(1):74–83, January 2002.
16. K. Kobara and H. Imai. “On the one-wayness against chosen-plaintext attacks on the Loidreau’s modified McEliece PKC”. *IEEE Trans. on IT*, 49(12), 2003.
17. K. Kobara, K. Morozov, and R. Overbeck. “Coding-based oblivious transfer”. In *Proc. of Mathematical Methods in Computer Science, LNCS 5393*, pages 142–156. Springer–Verlag, 2008.
18. P. Loidreau. “Strengthening McEliece cryptosystem”. In *Proc. of ASIACRYPT 2000*, pages 585–598. Springer–Verlag, 2000.
19. F. J. MacWilliams and N. J. A. Sloane. *“The theory of error-correcting codes”*, chapter 12, Sec. 3, Pr. 5. North-Holland Mathematical Library, 1977.
20. R. Misoczki and P. Barreto. personal communication, 2009.

21. R. Misoczki and P. Barreto. “Compact McEliece keys from Goppa codes”. In *Proc. of SAC ’09, LNCS 5867*, pages –. Springer–Verlag, 2009.
22. C. Monico, J. Rosenthal, and A. Shokrollahi. “Using low density parity check codes in the McEliece cryptosystem”. In *Proc. of IEEE International Symposium on Information Theory, ISIT ’00*, page 215, 2000.
23. R. Nojima, H. Imai, K. Kobara, and K. Morozov. “Semantic security for the McEliece cryptosystem without random oracles”. In *Proc. of WCC’07*, pages 257–268, 2007.
24. P.W. Shor. “Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer”. *SIAM Journal on Computing*, 26(5):1484–1509, 1997.
25. V. Shoup. “OAEP reconsidered”. In *Proc. of CRYPTO 2001*, pages 239–259, 2001.
26. J. Stern. “A new identification scheme based on syndrome decoding”. In *Proc. of CRYPTO ’93, LNCS 773*, pages 13–21. Springer–Verlag, 1994.
27. K. K. Tzeng and K. Zimmermann. “On extending Goppa codes to cyclic codes”. *IEEE Trans. on IT*, 21(6), 1975.
28. V. G. Umana and G. Leander. “Practical key recovery attacks on two McEliece variants”. `http://eprint.iacr.org/2009/509`, 2009.