A p-ary MDPC scheme

Qian Guo
Electrical and Information Technology
Lund University
Lund, Sweden
Email: qian.guo@eit.lth.se

Thomas Johansson
Electrical and Information Technology
Lund University
Lund, Sweden
Email: thomas.johansson@eit.lth.se

***Abstract*—The McEliece public key cryptosystem is an attractive general construction that has received extensive attention over the years. Recently, a very promising version called QC-MDPC, was proposed. By using binary quasi-cyclic codes, the size of the public key can be decreased significantly. The decryption step involves iterative decoding of moderate density parity check codes (MDPC). In this paper we propose a non-binary version of QC-MDPC. The errors in the new scheme are discrete Gaussian and the decryption involves a new type of iterative decoding with a non-binary alphabet. The resulting scheme improves upon the binary QC-MDPC in that the size of the pubic key can be even smaller.**

I. Introduction

Cryptosystems based on the hardness of factoring or discrete logarithm will be broken if a quantum computer is available [1]. Most of today's public-key cryptosystems used in practice, such as RSA or DSA, will thus be broken in such an event. Code-based cryptography is, on the opposite, believed to be quantum resistant and considered as an option for future applications.

A. Related Works

The McEliece cryptosystem [2] is the first code-based cryptosystem, originally proposed using Goppa codes. Although its implementation is efficient, it suffers from a very large key-size. A lot of proposals have been made to replace the originally proposed Goppa code family with other code families and many of them have failed. In particular, attempts have been made to considerably reduce the key-size, e.g. by using codes with a large automorphism group, such as quasi-cyclic codes. This has proved to be difficult due to the strong structure of such codes, giving security weaknesses.

However, in 2013 the QC-MDPC scheme was proposed [3] and this is today the most attractive McEliece type cryptosystem. The QC-MDPC scheme uses a family of simple quasi-cyclic codes which are of MDPC type. The parity checks in an MDPC code are similar to an LDPC code, with the difference that the weight of a parity checks in an MDPC code is not as low. The decryption step uses iterative decoding techniques, which in its simplest form can be Gallager's bit flipping algorithm [4]. The QC-MDPC scheme has a simple algebraic description and comes with some security reductions. As the quasi-cyclic structure allows the generator matrix to be reconstructed from a single matrix row, the key-size is significantly smaller than other schemes.

As a concrete parameter proposal for 80-bit security, the inventors of the QC-MDPC scheme used a rate $1/2$ binary code of length 9602, consisting of two cyclic matrices with row (or column) weight 45 each. In the encryption step an error of weight 84 is added. With a public generator matrix in systematic form, the key-size is 4801 bits.

Related to our work is also the NTRU cryptosystem [5]. NTRU is a public key encryption scheme in lattice-based cryptography. Similarities between QC-MDPC and NTRU have been mentioned previously, but one of several differences is that NTRU does not use iterative decoding techniques.

B. Contribution

In this paper we extend the QC-MDPC scheme from the binary field to a larger p-ary field. In the new scheme, the errors are drawn from a discrete Gaussian distribution or something similar and the decryption step involves a new type of iterative decoding with a non-binary alphabet. The resulting scheme improves upon the binary QC-MDPC in that the size of the pubic key can be even smaller, but still the complexity of the iterative decoding step is kept small.

The remaining parts of the paper are organized as follows. We give some preliminaries on coding theory in Section II, and then state general and specific proposals of the new p-ary MDPC McEliece scheme in Section III and IV, respectively. Section V presents the new iterative decoder. This is followed by a security assessment part in Section VI and a discussion part in Section VII. We finally conclude the paper in Section VIII.

II. Preliminaries

We present some basic concepts in coding theory. Let $\mathbb{F}_p$ denote a finite field of a prime size p.

Definition 1 (Linear Codes): An $[n, n-r]$ linear code $\mathcal{C}$ over a field $\mathbb{F}_p$ is an $(n-r)$-dimensional vector subspace of $\mathbb{F}_p^n$. Its co-dimension is r, characterizing the redundancy of the code.

The authors are supported by the Swedish Research Council (Grants No. 2015-04528). Qian Guo is also supported by an Erasmus Mundus Scholarship. This is the author's version of the paper "A p-ary MDPC scheme", which was published in the proceedings of the IEEE International Symposium on Information Theory (ISIT) 2016. The final, published version is available through IEEE Xplore at doi:10.1109/ISIT.2016.7541520. Minor differences may exist between this version and the version of record due to copyediting and publisher formatting.

A generator matrix $\mathbf{G}$ of the linear code $\mathcal{C}$ is defined as an $(n-r)\times n$ matrix in $\mathbb{F}_p^{(n-r)\times n}$ whose rows form a basis of the code. Equivalently, the codes can be defined by a matrix $\mathbf{H}$ in $\mathbb{F}_p^{r\times n}$ whose kernel is the code $\mathcal{C}$, called a parity-check matrix of $\mathcal{C}$. We note that in most cases, both the generator and parity-check matrices are not unique, but the linear code $\mathcal{C}$ has a unique counterpart, a dual code $\mathcal{C}^{\perp}$ spanned by the rows of one of its parity-check matrices.

Definition 2 (Quasi-cyclic Codes): Suppose $n = n_0 r$. An $[n, r]$-linear code $\mathcal{C}$ over $\mathbb{F}_p$ is quasi-cyclic if every cyclic shift of a codeword by n_0 places remains a codeword.

We can conveniently represent both the generator and parity-check matrices by a series of $r \times r$ circulant blocks. Thus, each block is determined by its first row. The advantage of this representation comes from the isomorphism between the algebra of these matrices and that of polynomials modulo $x^r - 1$ over the field $\mathbb{F}_p$, which provides both computational efficiency and security guarantee for several carefully-chosen parameters of r.

Definition 3 (LDPC/MDPC Codes): A low density parity-check code (LDPC) is a linear code admitting a sparse parity-check matrix, while an MDPC code is a linear code with a denser but still sparse parity-check matrix.

In the previous works on binary LDPC/MDPC, or p-ary LDPC, the Hamming weight of the row vector, i.e., the number of its non-zero component, is usually employed to characterize its sparsity: LDPC codes are with small constant row weights; MDPC codes with row weights scale in $O(\sqrt{n \log n})$. In this work, we will use the Euclidean metric, and specify a sparse parity-check matrix to be one with certain structures.

III. The New Scheme—A General Description

In this section, we describe the general scheme and the underlying code constructions. Note that since the new proposal is an extension, its main structure is similar to that of the binary (QC)-MDPC scheme [3]. We will state the distinctions in Section VII-A.

A. A p-ary MDPC Code Construction

- Generate r vectors $(\mathbf{h_i} \in \mathbb{F}_p^n)_{0\le i<r}$, each with w_{sig} significant entries, w_1 entries chosen from $\{-1, 1\}$, w_2 entries from $\{-2, 2\}$, and the remaining set to 0.
- The parity-check matrix $\mathbf{H} \in \mathbb{F}_p^{r\times n}$ of the i^{th} row $\mathbf{h_i}$ defines its corresponding p-ary MDPC Code.

B. A p-ary quasi-cyclic MDPC Code Construction

Let $n = n_0 r$.

- Generate a vector $\mathbf{h} \in \mathbb{F}_p^n$, each with w_{sig} significant entries, w_1 entries chosen from $\{-1, 1\}$, w_2 entries from $\{-2, 2\}$, and the remaining set to 0.
- The parity-check matrix $\mathbf{H} \in \mathbb{F}_p^{r\times n}$ with the first row $\mathbf{h}$ defines its corresponding p-ary QC-MDPC Code—the other $r-1$ rows are just the $r-1$ quasi-cyclic shifts of $\mathbf{h}$.

Because of the quasi-cyclic feature, we construct a parity-check matrix $\mathbf{H} = [\mathbf{H_0}|\mathbf{H_1}|\dots|\mathbf{H_{n_0-1}}] \in \mathbb{F}_p^{r\times n}$, where each of its blocks is isomorphic to a polynomial

$$h_i(x) \in R = \mathbb{F}_p[x]/\langle x^r - 1\rangle.$$

Using classic methods (e.g., [6]) on coding theory, we generate a dense generator matrix determined by polynomials in the same residue ring. The number of significant positions in each block is denoted by $w_{\text{sig},i}$, and is called its sig-weight. Thus, $w_{\text{sig}} = \sum_{i=0}^{n_0-1} w_{\text{sig},i}$.

We recommend to set r to be a prime number[1], just as some proposals in NTRU [5]. This security guarantee is intuitive, and we leave a rigorous reduction to one of the well-studied hard lattice problems (e.g., SVP) as a future work.

C. The Scheme

- KeyGen():
  - Generate a parity-check matrix $\mathbf{H}$ with the required special properties.
  - Derive its corresponding generator matrix $\mathbf{G}$ in row reduced echelon form. Here $\mathbf{G}$ should be a dense matrix; otherwise, the parity-check matrix $\mathbf{H}$ should be regenerated.
  - The public key: $\mathbf{G}$.
    The private key: $\mathbf{H}$.
- $\text{Enc}_{\mathbf{G}}(\mathbf{m})$:
  - Generate a random vector $\mathbf{e}$. It is usually generated according to a discrete Gaussian distribution, but as we can see later, sometimes other easy-implemented distributions will also be employed.
  - The ciphertext is $\mathbf{c} = \mathbf{mG} + \mathbf{e}$.
- $\text{Dec}_{\mathbf{H}}(\mathbf{c})$:
  - Compute the syndrome vector $\mathbf{s} = \mathbf{cH}^{\text{T}} = \mathbf{eH}^{\text{T}}$, and then use an iterative decoder to extract the noise $\mathbf{e}$.
  - Recover the plaintext $\mathbf{m}$ from the first $(n-r)$ entries of $\mathbf{mG}$.

Notice that similar to the descriptions in [3], we exclude the scrambling matrix $\mathbf{S}$ and permutation matrix $\mathbf{P}$ because of the assumption that the generator matrix $\mathbf{G}$ is a dense matrix. Moreover, making use of the CCA-2 security-conversion, e.g. [9], we can represent the generator matrix $\mathbf{G}$ in a systematic form, thereby reducing the public-key size of the proposed p-ary QC-MDPC scheme to $(n-r)\log_2 p$.

IV. A Specific Proposal

In this section, we detail a specific p-ary QC-MDPC McEliece proposal by making several simplifying assumptions.

- First, we generate the noise vector $\mathbf{e}$ according to an alternative distribution, i.e., the uniform distribution in

[1]We can also use p-ary quasi-negacyclic codes, and set r to be 2^u, where u is a natural number. In this case, we work on a ring $\mathbb{F}_p[x]/\langle x^r + 1\rangle$, which is a safer ring as recommended in RING-LWE-based cryptography [7][8]. However, there are fewer parameter choices as r is set to be an integer 2^u.

$\{0, -1, 1\}^*$, and take this setting throughout the remaining paper.

- Second, for ease of controlling the short cycles of the corresponding Tanner Graph, we assume that the significant positions are distributed evenly in each block of the sparse p-ary QC-MDPC parity-check matrix. That is, the row sig-weights $w_{\text{sig},i}$'s differ by at most 1.
- Third, for ease of decoding, we assume that the most significant entry h_{i_1} is uniformly chosen from $[\frac{p}{2}-\delta_1, \frac{p}{2}+\delta_1]$, ..., and the $w_{\text{sig}}^{\text{th}}$ significant entry $h_{i_{\text{sig}}}$ uniformly chosen from $[\frac{p}{2^{w_{\text{sig}}}} - \delta_{w_{\text{sig}}}, \frac{p}{2^{w_{\text{sig}}}} + \delta_{w_{\text{sig}}}]$. Here $(\delta_i)_{1\le i\le w_{\text{sig}}}$ are algorithmic parameters chosen by concrete settings.

NOTE: Using this setting, we can limit each entry of $\mathbf{eH}^{\text{T}}$ to lie in the interval[2] $(-\frac{3p}{2}, \frac{3p}{2})$, when the operations are viewed over $\mathbb{R}$. The combinations of the significant coefficients are several integer points far away from each other. If the noise variance is well-controlled, then we can always reach the correct combination and at least decode some values correspond to the combination — this is an intuitive description why our decoder works.

V. Decoder

In this section, we present a new iterative hard-decision decoder for the proposed p-ary QC-MDPC code. This decoder consists of several rounds of passing soft-information twice and can be modified to succeed with probability close to 1 using some heuristic assumptions, which is vital for a cryptosystem. Another interesting observation is that the proposed decoder outperforms its soft-decision counterpart in simulation.

A. The Corresponding Tanner Graph

Similar to that in the iterative decoding of binary LDPC/MDPC codes, we need to first build the bipartite Tanner graph corresponding to the parity-check matrix $\mathbf{H}$. This extension is not straight-forward for the new p-ary codes in the Euclidean metric, since the corresponding Tanner graph will be quite dense if it is build in a normal (Hamming) manner, which is a major obstacle for an efficient decoding.

The novel solution is to form a sparse Tanner graph by keeping the edges with a significant coefficient and adding in each check node a new edge connecting to an error node representing the contribution of the edges with a small coefficient. The apriori probability of the value in each error node can be pre-computed and stored in a table, allowing an efficient hard-decision version in the later subsection. We can also employ this sparse graph to perform the classic sum-product algorithm via fully updating the distribution of the value in the imaginary error node.

Moreover, when designing the system, the user should avoid choosing from the key space a secret parity-check matrix that will introduce short cycles in its corresponding Tanner graph. We can ensure that its Tanner graph is cycle-free for the proposed parameter setting in Section VI-A.

[2]A more general setting is making sure that each entry lies in $(-\frac{p}{2} - |a|p, \frac{p}{2}+|a|p)$, where a is an integer with a small absolute value. The deduced decoder is also applicable.

B. A Hard Decoding Strategy

We give a brief description of the proposed hard-decision decoder here due to the page limit and refer the interested readers to the full version [10] for details.

- Start with the initial parity-check matrix $\mathbf{H}^{(0)} = \mathbf{H}$, and the initial syndrome $\mathbf{s}^{(0)} = \mathbf{s}$.
- For the t^{th} iteration:
  - Set the distribution of each undetermined message node to be uniform over $\{-1, 0, 1\}$ and the distribution of the imaginary error node corresponds to the check node v_j to be the apriori distribution $D_j^{(t)}$, which is pre-computed and stored in a table.
  - Perform two rounds of the classic sum-product algorithm. Note that since the degree of the imaginary error node is 1, its corresponding distribution is unchanged during the two-pass message-passing process.
  - In the message node X_j, if its entropy is rather small, i.e., with a probability[3] larger than $1-\epsilon$, the variable X_j is equal to a certain value $x_j \in \{-1, 0, 1\}$, we set X_j to be x_j. We then update $\mathbf{H}^{(t)}$ by removing the columns in the current parity-check matrix $\mathbf{H}^{(t-1)}$ whose corresponding message value is determined, and also re-compute the syndrome $\mathbf{s}^{(t)}$ by substituting the determined values x_j's.
- Terminate if reaching the limit on the maximum number of iterations. If the number of undetermined message nodes is less than a pre-set threshold, perform Gaussian Elimination to recover these message values; report failure otherwise.
- Check all the parity-check equations, and report failure if one is unsatisfied.

C. Treating the Decoding Error Probability

We have implemented both the above hard-decision decoder and its soft-decision counterpart and obtain satisfying performance. For example, if we use the parameter setting proposed for 80-bit security in Section VI-A, the word error probability after 4 iterations is only 3×10^{-6}. Moreover, the hard-decision one can handle a much noisier distribution in the imaginary error nodes compared with its soft-decision counterpart.

Another key issue is to treat its non-zero decoding error probability, which can be solved via simply making use of the similar methods as in [3]. For this new hard-decision decoder, we introduce a heuristic variant to make the error probability small by additionally calling the decoder a constant number of times.

The procedure works as follows. When a decoding failure is reported, we re-perform the first iteration of the hard-decoding process; we then randomly choose a fraction of

[3]Here ϵ is extremely small; for example, we set it to be 10^{-9} in our implementation.

the determined values, update the parity-check matrix and the syndrome vector according to these selected values, and call the decoder again; we repeat this choosing-updating-decoding procedure until the decoding succeeds or reaching the limit on the number of iterations.

Using this approach, we can make the decoding failure undetectable in several million decoding tests and show that under some heuristic assumptions the decoding error probability can be reduced to less than 2^{-80} (or even much smaller) for the given parameter setting (See [10]).

D. The Complexity Analysis

Since the code length is smaller and the corresponding Tanner graph is sparser compared with the binary-MDPC scheme, the decoding complexity of the new scheme is competitive even if some soft information is used during the message-passing process. To be specific, the proposed instantiation in Section VI-A for 80-bit security will require far less operations (less than 20%) than its binary-MDPC counterpart.

VI. Possible Attacks

While the best technique for solving the binary McEliece schemes generally is still information set decoding (ISD) [11][12], it works poorly for the newly proposed p-ary MDPC scheme with Euclidean noise. Therefore, it is promising to assess the security levels, by using techniques designed for attacking lattice-based cryptography that employs the Euclidean metric. We estimate its security against both message-recovery attacks and key-recovery attacks.

For a message-recovery attack, we are facing an exact (Ring-) LWE [13][7] (or (Ring-) LWE with small errors) problem with dimension $(n-r)$ and a uniform noise distribution in $\{-1,0,1\}$, whose hardness with a limited sample number has been ascertained in [14] by a reduction to some hard lattice problems. Note that for distinct instances, the most competitive attacks (e.g, combinatorial attacks like BKW [15] or Meet-In-the-Middle (MITM), lattice attacks like SIS sieving or Bounded Distance Decoding, the hybrid attack [16] for NTRU, and other algebraic attacks like Arora-Ge [17] and Gröbner based attacks [18]) are distinct[4]. Thus, every known attack should be tested to ensure that the concrete complexity of the instance is larger than its designed security level.

For a key-recovery attack, the problem equals that of finding a vector $\mathbf{h}$ with special structures, i.e., several entries are significant and the remaining part is short, such that

$$\mathbf{G}\mathbf{h}^{\mathrm{T}} = \mathbf{0},$$

where $\mathbf{G}$ is the public key, a dense generator matrix. This is exactly the content of finding a special codeword in the dual code $\mathcal{C}^{\perp}$, but in the Euclidean metric sense. In the literature of lattice-based cryptography, it has another name—Short Integer Solution (SIS), and can be solved using methods similar to those for LWE.

[4]As the sample number is limited, the algebraic-type attacks cannot succeed in polynomial time and the BKW-type attacks need to generate new samples from these given ones.

In the p-ary QC-MDPC case, more security aspects should be considered due to its additional algebraic structures. In particular, we should choose the parameters to resist a general attack that reduces the instance to another with a much smaller dimension and still controllable uncertainty, therefore breaking some instantiations of NTRU [19], Ring-LPN [20] or McEliece cryptosystems [21][22]. As recommended in Section III-B, we choose r to be a prime integer.

A. An Instance for the 80-bit Security

In this part, we propose a simple instantiation of the new p-ary quasi-cyclic-MDPC McEliece crypto-system for achieving the 80-bit security. The parameters are as follows,

$$n = 614, p \text{ a prime} \approx 2^{10}, r = 307, w_{\text{sig}} = 4, w_1 = 80, w_2 = 6.$$

Thus, the chosen parity-check matrix $\mathbf{H}$ has two blocks, and the proposed system is close to NTRU. We also set all $(\delta_i)_{i\in\{1,\ldots,4\}}$ to be 3 to further increase the key space.

The above instantiation corresponds to a Tanner graph with a quite simple structure, and also provides an extremely large key space. In addition, in each block, if the interval length between two significant entries in a row is co-prime to p, then the corresponding Tanner graph is cycle-free. Thus, we remove all the unwanted parity-check matrices that form Tanner graphs with cycles, to ensure good decoding performance.

To the best of our knowledge, this instance thwarts all known attacks with computational power limited to 2^{80} bit-operations. On one side, we need to solve an LWE instance with dimension 307, field size about 2^{10}, a uniform noise distribution in $\{-1,0,1\}$, and a limited sample size, to form a message-recovery attack. The recent best combinatorial (BKW-type) solvers [23][24] or pure lattice-reduction-based solvers will cost more bit-operations compared with the hybrid attack, which is also the most promising attack on NTRU and requires more than 2^{80} bit-operations according to the recent analysis in [25].

On the other side, we argue that a successful key-recovery attack is an even more challenging task. First, the four significant entries will ruin all the known attacks searching for short vectors in the space of the dual code $\mathcal{C}^{\perp}$, as they make the length of the secret vector really large; even if their influence could be removed costlessly, then the remaining problem is still hard in the corresponding security level, because the dimension is not highly reduced. Note that for a key-recovery attack, the hybrid attack is inefficient owing to the several entries from $\{2,-2\}$ introduced in the private matrix $\mathbf{H}$.

We see that its key-size is approximately 3070 bits, less than two-thirds compared with the cryptosystem based on binary MDPC codes [3]. We will add more instantiations for different security levels in the full version of the paper.

VII. Discussion

This new p-ary QC-MDPC McEliece scheme shares some similarities with its binary counterpart QC-MDPC and also with the NTRU scheme. This is just an incipient attempt to combine both the iterative decoding technique and the use of

more compact way for representing information. This scheme is very attractive as a mixture of topics in both lattice-based cryptography and code-based cryptography. In this section, we pay more attentions to the variations.

A. The Comparison with Binary QC-MDPC

The direct differences to binary QC-MDPC are that the new scheme is p-ary and uses the Euclidean metric, which force us to redesign the decoder. These new features allow us to resist the ISD attack easily and change the major threat to other attacks.

The main advantage of the new p-ary QC-MDPC McEliece scheme is its extremely compact keys. This makes it a stimulating research topic, as it further improves the major drawback of the well-known McEliece cryptosystem significantly, along the path of binary QC-MDPC.

B. The Comparison with NTRU

NTRU is a commercialized cryptosystem, and has been analyzed for more than 15 years. Compared with NTRU, therefore, the new scheme still has a long way to go. The following concludes their main differences.

- First, in NTRU, the message is a sparse polynomial, while the new scheme may offer a larger message space.
- In addition, the new scheme will choose a longer secret vector, offering more security for the attacks against NTRU that search for a short vector. Moreover, several entries chosen from $\{2, -2\}$ in the private matrix are used to protect against the hybrid attacks.
- Last but not least, the new scheme exploits the iterative decoding technique to provide some favorable features. For example, in NTRU, folding is prohibited, since the decryption algorithm will fail if the coefficients of a polynomial generated during decryption do not lie in an interval with width p; using iterative decoding, we can handle a larger interval. Thus, it is sufficient to choose a smaller prime p as the underlying field size, thereby both reducing the key-size and improving the invulnerability against lattice attacks.

VIII. Conclusions and Future Works

In this paper, we have presented a new p-ary MDPC McEliece cryptosystem, which extends the recent binary MDPC scheme [3] and employs the Euclidean metric to bound noise. Besides, using a quasi-cyclic structure, the key is extremely compact—for 80-bit security, the key-size of one instantiation is less than two-thirds compared with its binary counterpart. Taking all these modifications into consideration, we have also presented a hard-decision iterative decoder well controlling the decoding complexity.

There are a few obvious improvements, one being to design a new decoding strategy to further improve the trade-off between decoding complexity and performance, another being to find a more efficient implementation (e.g., to borrow ideas [26] from lattice-based cryptography). For the proposed instantiation, its security is mainly assessed by recent results on cryptanalysis, so another interesting direction is to find a tight reduction to some hard ideal lattice problems.

References

[1] P. W. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," in *35th Annual Symposium on Foundations of Computer Science, 20-22 November 1994, Santa Fe, New Mexico, USA*. IEEE Press, 1994, pp. 124–134.

[2] R. J. McEliece, "A public-key cryptosystem based on algebraic coding theory," *DSN Progress Report 42–44*, pp. 114–116, 1978.

[3] R. Misoczki, J.-P. Tillich, N. Sendrier, and P. S. Barreto, "MDPC-McEliece: New McEliece variants from moderate density parity-check codes," in *Information Theory Proceedings (ISIT), 2013 IEEE International Symposium on*. IEEE, 2013, pp. 2069–2073.

[4] R. G. Gallager, "Low-Density Parity-Check Codes," Ph.D. dissertation, MIT Press, Cambridge, 1963.

[5] J. Hoffstein, J. Pipher, and J. H. Silverman, "Ntru: A ring-based public key cryptosystem," in *Algorithmic number theory*. Springer, 1998, pp. 267–288.

[6] R. Roth, *Introduction to coding theory*. Cambridge University Press, 2006.

[7] V. Lyubashevsky, C. Peikert, and O. Regev, "On ideal lattices and learning with errors over rings," *Journal of the ACM (JACM)*, vol. 60, no. 6, p. 43, 2013.

[8] ——, "A toolkit for ring-lwe cryptography," in *Advances in Cryptology–EUROCRYPT 2013*. Springer, 2013, pp. 35–54.

[9] K. Kobara and H. Imai, "Semantically secure mceliece public-key cryptosystems-conversions for mceliece pkc," in *Public Key Cryptography*. Springer, 2001, pp. 19–35.

[10] Q. Guo and T. Johansson, "A p-ary MDPC scheme (Full version)," in *In preparation*, 2015.

[11] J. Stern, "A method for finding codewords of small weight," in *Coding Theory and Applications, 3rd International Colloquium, Toulon, France, November 2-4, 1988, Proceedings*, 1988, pp. 106–113.

[12] A. Becker, A. Joux, A. May, and A. Meurer, "Decoding random binary linear codes in $2^{n/20}$: How 1 + 1 = 0 improves information set decoding," in *Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings*, 2012, pp. 520–536. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-29011-4_31

[13] O. Regev, "On Lattices, Learning with Errors, Random Linear Codes, and Cryptography," *Journal of the ACM*, vol. 56, no. 6, pp. 34:1–34:40, Sep. 2009. [Online]. Available: http://doi.acm.org/10.1145/1568318.1568324

[14] D. Micciancio and C. Peikert, "Hardness of SIS and LWE with Small Parameters," in *Advances in Cryptology–CRYPTO 2013*. Springer, 2013, pp. 21–39.

[15] A. Blum, A. Kalai, and H. Wasserman, "Noise-tolerant learning, the parity problem, and the statistical query model," *J. ACM*, vol. 50, no. 4, pp. 506–519, 2003.

[16] N. Howgrave-Graham, "A hybrid lattice-reduction and meet-in-the-middle attack against ntru," in *Advances in Cryptology-CRYPTO 2007*. Springer, 2007, pp. 150–169.

[17] S. Arora and R. Ge, "New algorithms for learning in presence of errors," in *Automata, Languages and Programming*. Springer, 2011, pp. 403–415.

[18] M. Albrecht, C. Cid, J.-C. Faugere, F. Robert, and L. Perret, "Algebraic algorithms for lwe problems," *Cryptology ePrint Archive, Report 2014/1018*, 2014.

[19] C. Gentry, "Key recovery and message attacks on NTRU-composite," in *Advances in Cryptology–Eurocrypt 2001*. Springer, 2001, pp. 182–194.

[20] Q. Guo, T. Johansson, and C. Löndahl, "A new algorithm for solving Ring-LPN with a reducible polynomial," *IEEE Transactions on Information Theory*, vol. 61, no. 11, pp. 6204–6212, 2015.

[21] C. Löndahl, T. Johansson, M. K. Shooshtari, M. Ahmadian-Attari, and M. R. Aref, "Squaring attacks on mceliece public-key cryptosystems using quasi-cyclic codes of even dimension," *Designs, Codes and Cryptography*, pp. 1–19.

[22] J. Faugère, A. Otmani, L. Perret, F. de Portzamparc, and J. Tillich, "Folding Alternant and Goppa Codes With Non-Trivial Automorphism Groups," *IEEE Trans. Information Theory*, vol. 62, no. 1, pp. 184–198, 2016.

[23] Q. Guo, T. Johansson, and P. Stankovski, "Coded-bkw: Solving lwe using lattice codes," in *Advances in Cryptology–CRYPTO 2015*. Springer, 2015, pp. 23–42.
[24] P. Kirchner and P.-A. Fouque, "An improved bkw algorithm for lwe with applications to cryptography and lattices," in *Advances in Cryptology–CRYPTO 2015*. Springer, 2015, pp. 43–62.
[25] J. Hoffstein, J. Pipher, J. M. Schanck, J. H. Silverman, W. Whyte, and Z. Zhang, "Choosing parameters for ntruencrypt," Tech. Rep.
[26] R. De Clercq, S. S. Roy, F. Vercauteren, and I. Verbauwhede, "Efficient Software Implementation of Ring-LWE Encryption," *Design, Automation and Test in Europe (DATE 2015)*, 2015.