Code-Based Cryptography:

State of the Art and Perspectives

Nicolas Sendrier | INRIA

Code-based cryptography is one of the few mathematical techniques that enables the construction of public-key cryptosystems that are secure against an adversary equipped with a quantum computer. The McEliece public-key encryption scheme and its variants are candidates for a postquantum public-key encryption standard.

Code-based cryptography is one of the main postquantum techniques available, together with lattice-based cryptography, multivariate cryptography, and hash-based cryptography. Robert McEliece proposed the first code-based cryptosystem in 1978.[1] It belongs to a very narrow class of public-key primitives that have resisted all cryptanalytic attempts up to now.

In this survey, I present the McEliece public-key encryption scheme's basic principles and security assumptions,[2] illustrating that the system is secure and practical despite its large key size. I also consider McEliece variants with compact keys; using quasicyclic codes, it's possible to design schemes with much shorter keys. I review the principle of those constructions with a particular focus on quasicyclic moderate density parity check (MDPC) codes, which combine a good security proof with much shorter public keys, allowing a simple and efficient key exchange protocol with forward secrecy.

McEliece Scheme and Its Main Variants

The McEliece public-key encryption scheme was proposed almost 40 years ago and hasn't been threatened essentially since then. McEliece's original idea was to use as ciphertext a word of a carefully chosen linear error-correcting code—a binary Goppa code, in this case—to which random errors were added.[1] An arbitrary basis of the code—a generator matrix—is the public key, allowing anyone to encrypt (see Figure 1). Legitimate users who know a secret trapdoor—a fast (that is, polynomial time) decoding algorithm for the code—can remove the errors and recover the cleartext. Adversaries are reduced to a generic decoding problem, which is believed to be hard on average, including against quantum adversaries.

Security Assumptions

The scheme's security relies on two computational assumptions:

- generic decoding is hard on average, and
- the public key—a generator matrix—is hard to distinguish from a random matrix.

By *hard*, I mean that the underlying computational problem can't be efficiently solved, or more to the

 Copublished by the IEEE Computer and Reliability Societies

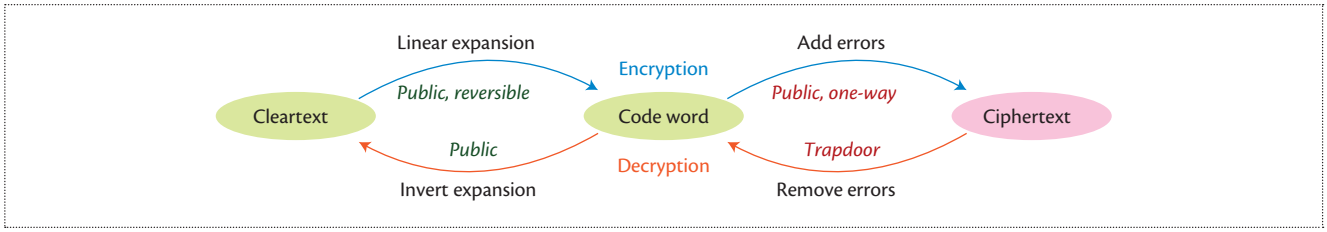

Figure 1. Code-based public-key encryption. The ciphertext is a noisy code word that only the legitimate user can correct to recover the cleartext.

point, by choosing appropriately large sizes, those problems are intractable and thus, under these assumptions, the system is secure against any computationally bounded adversary.

The twofold security proof has been implicitly understood since the origin of the scheme but was first stated in Nicolas Courtois and his colleagues' "How to Achieve a McEliece-Based Digital Signature Scheme"[3] and formally proven in my paper "On the Use of Structured Codes in Code Based Cryptography."[4] The first problem, generic decoding, is nondeterministic polynomial time (NP)-complete[2] and is also believed to be hard on average.[5] Progress is possible and would then require an increase in system parameters, but a significant breakthrough is unlikely. Much like factoring and discrete logarithms for number theory–based cryptosystems, research on this topic must be maintained at the highest level to ensure enough confidence in the system and adjust its parameters when needed.

The second problem, public-key indistinguishability, is much more open. To state it properly, the system must be instantiated. For instance, McEliece proposed using the family of binary Goppa codes, for which the indistinguishability assumption holds so far (except when the code rate tends to 1,[6] which is an irrelevant case for encryption). For some other families, Reed-Solomon codes, concatenated codes, low-density parity check codes, and so on, the assumption doesn't hold and the corresponding instances of McEliece are unsafe. Providing families of codes for which the indistinguishability assumption holds is a key issue in code-based cryptography.

Figure 2 gives a general description of the McEliece encryption scheme. Terminology refers to the coding theory glossary in the sidebar.

Other Cryptographic Primitives from Codes

Public-key encryption can also be achieved with the Niederreiter scheme,[7] which is equivalent to the McEliece scheme in terms of security. In addition, two other important functionalities can be achieved from codes: zero-knowledge authentication and digital signatures.

Parameters:	A family $\mathcal{F}$ of binary linear t-error correcting $[n,k]$ codes	
Key generation:	$\rightarrow (G, \Phi)$ a key pair where	
public key:	$G \in \{0,1\}^{k \times n}$ is a matrix which spans a code $\langle G \rangle \in \mathcal{F}$	
private key:	Φ is a t-bounded decoder for $\langle G \rangle$	
Encryption:	$\{0,1\}^k \rightarrow$	$\{0,1\}^n$
	$x \mapsto$	$xG + e$ with $e \in \{0,1\}^n$ random of Hamming weight t
Decryption:	$\{0,1\}^n \rightarrow$	$\{0,1\}^k$
	$y \mapsto$	$\Phi(y)G^*$ with G^* a right inverse of G

Figure 2. McEliece public-key encryption scheme.

Zero-knowledge authentication protocol. The first such protocol was proposed by Jacques Stern in 1993.[8] Some variants have followed, and all amount to the same idea: one party picks a code word x, keeps it secret, and publishes a noisy version of it, say $y = x + e$, with e of small weight. Then, this party can prove interactively to another party that it knows a code word close to the public word y without ever revealing any information about x.

Digital signature. There's a generic way to produce digital-signature schemes from zero-knowledge protocols using the Fiat-Shamir paradigm.[9] This can be achieved using the Stern protocol. Note that, against quantum adversaries, the construction requires some modifications.[10] The resulting digital signature scheme is easy to implement and enjoys relatively small key sizes (a few hundred bytes) but produces rather large signatures (one or a few hundred kilobytes).

Another method to build digital signatures is the "hash and sign" paradigm in which users consider the digest of the message to be signed as a ciphertext and produce the corresponding cleartext as the signature. In this scenario, the public key can be used to check the signature's validity. Unfortunately, the McEliece encryption primitive isn't surjective and the parameters used for encrypting aren't suitable for signing. Digital signature is nevertheless possible but requires families of binary Goppa codes of a rate close to 1,[3] precisely the subclass of Goppa codes that is distinguishable from random.[6] Even though this distinguisher doesn't lead to an effective attack, it invalidates the

A Coding Theory Glossary

I consider here only linear codes over the binary field $\mathbf{F}_2 = \{0,1\}$ of integers modulo 2. Codes over larger alphabets, any finite field $\mathbf{F}_q$ with q elements, can be defined likewise and are sometimes used in cryptography. Most statements and claims are valid in general. Nonlinear codes also exist but are beyond this survey's scope.

Hamming distance: The distance between two words x and y of same length is the number of coordinates in which they differ, denoted dist(x,y).

Hamming weight: The weight of a word x determined by its number of nonzero coordinates, denoted $wt(x)$.

Linear code: A binary linear $[n,k]$ code C of dimension k and length $n \geq k$ is a k-dimensional subspace of the vector space $\{0,1\}^n$. An element of a code is a code word. The code rate is the ratio k/n.

Generator matrix: A full-rank $k \times n$ matrix $G \in \{0,1\}^{k\times n}$ whose rows form a basis of C. $C = \{xG \mid x \in \{0,1\}^k\}$. The mapping $x \mapsto xG$ expands any k-bit word into an n-bit code word.

Systematic generator matrix: A $k \times n$ generator matrix is in systematic form if it contains a square $k \times k$ identity matrix, usually in its first k columns. The systematic form always exists and reduces the storage requirements from nk to $(n - k)k$ bits.

Parity check matrix: A full-rank $(n - k) \times n$ matrix $H \in \{0,1\}^{(n-k)\times n}$ whose rows are orthogonal to C. $C = \{y \in \{0,1\}^n \mid yH^T = 0\}$.

Bounded decoding: A t-bounded decoder for $C[n,k]$ is a mapping Φ: $\{0,1\}^n \rightarrow C$ such that for all y in $\{0,1\}^n$ and all x in C, dist(y,x) $\leq t$ implies $\Phi(y) = x$.

Code family: Among the issues and successes of coding theory, one was to design "good" families of codes. Typically, the problem was to find, for given but arbitrary large values of n and k, families of $[n,k]$ codes for which an efficient (that is, polynomial time) t-bounded decoder could be devised, where t is a function of n and k. The larger t, the better the code family.

For cryptographic purposes, a family F of t-error correcting linear $[n,k]$ codes is chosen to provide the desired security level. Choosing a code C in the family will naturally provide a pair (G,Φ) where G is a generator matrix of C and Φ is a t-bounded decoder for C.

Generic decoding: A generic decoder for binary linear $[n,k]$ codes is a mapping Ψ: $\{0,1\}^n \times \{0,1\}^{k\times n} \rightarrow \{0,1\}^k$. For a given instance $x = \Psi(y,G)$, a measure of success is the Hamming distance between xG and y, which must be as small as possible. In particular, a t-bounded generic decoder for $[n,k]$ codes will be such that for all $x \in \{0,1\}^k$, for all $G \in \{0,1\}^{k\times n}$, and all $e \in \{0,1\}^n$ of Hamming weight t or less, we have $\Psi(xG + e,G) = x$. Generic decoding is nondeterministic polynomial time (NP)-hard[1] and difficult on average;[2] the best known solution's complexity grows exponentially with error weight t.[3]

References

1. E.R. Berlekamp et al., "On the Inherent Intractability of Certain Coding Problems," *IEEE Trans. Information Theory*, vol. 24, no. 3, 1978, pp. 384–386.
2. M. Alekhnovich, "More on Average Case vs Approximation Complexity," *Proc. 44th Symp. Foundations of Computer Science* (FOCS 03), 2003, pp. 298–307.
3. A. May and I. Ozerov, "On Computing Nearest Neighbors with Applications to Decoding of Binary Linear Codes, *Advances in Cryptology* (EUROCRYPT 15), Part I, LNCS 9056, Springer, 2015, pp. 203–228.

security proof. Moreover, the scheme uses large public keys, has significant signing complexity, and doesn't scale very well.

The Practice

Here I address McEliece's semantic security, implementation, and parameter selection.

Semantic security. The McEliece encryption scheme is very malleable. For instance, adding a code word to a ciphertext produces the ciphertext of a cleartext different from the original one. Also, a cleartext that's encrypted twice with the same key is revealed.

Consequently, a semantically secure conversion is mandatory.[11] An interesting side effect of such a conversion is that having a public key in systematic form becomes harmless, whereas this was inadvisable in the original scheme. A systematic public key reduces the public key size.

Implementation. To my knowledge, there's no widely deployed cryptographic product using code-based primitives. Still, there's a practice, and researchers have reported various efficient implementations in software and for embedded devices. The current state of the art for software is McBits,[12] which is fully protected against timing attacks and performs as well—if not better—than other asymmetric schemes.

Parameter selection. To achieve "classical" security of 128 bits (the best-known attacks require at least 2^{128} elementary operations) with the original McEliece scheme

using binary Goppa codes, codes must be of dimension $k = 3{,}376$ probably and length $n = 4{,}096$ correcting $t = 60$ errors. The public key size is 303,840 bytes (with a matrix in systematic form), and the expansion between the cleartext and the ciphertext is approximately 20 percent. For the same security against quantum adversaries, the block size must increase by a factor of 2, and the key size by a factor of 4.

Pros and Cons

The pros of the McEliece scheme and its variants include the following:

- security is well understood,
- it has resisted 40 years of scrutiny, and
- it's computationally efficient in encryption and decryption.

Cons include the following:

- there's no really practical digital-signature scheme,
- the public key size is large (on the order of one megabyte for long-term security), and
- the code indistinguishability assumption needs to be explored further.

A New Trend: Compact Keys from Quasicyclic Codes

By choosing a proper code family, it's possible to restrict the choice of the public generator matrix G to block-circulant matrices (see Figure 3), allowing significant reduction of the public key's storage requirements. A code spanned by a block-circulant matrix is *quasicyclic*. Phillipe Gaborit first proposed such constructions,[13] and in "On the Use of Structured Codes in Code Based Cryptography," I proved that the security proof in essentially unchanged.[4] The system is secure as long as

- generic decoding in a quasicyclic code is hard, and
- the public key is indistinguishable from a random block-circulant matrix.

Much like for cyclic variants of lattice hard problems, there's a consensus that generic decoding for quasicyclic codes remains hard. On the other hand, quasicyclic codes are more structured, and the code family must be chosen carefully. Note also that compact keys can be obtained with other constructions, like block-dyadic matrices, with a similar effect on the key size and security.

Block-Circulant Matrices

A circulant matrix is a square matrix in which each row is the rotation one element to the right of the preceding row. It's completely defined by its first row. A block-circulant matrix is formed of circulant square blocks. The index of a block-circulant matrix comprises the number of circulant blocks in each row. Its order is the size of the circulant blocks. I extend this terminology to quasicyclic codes.

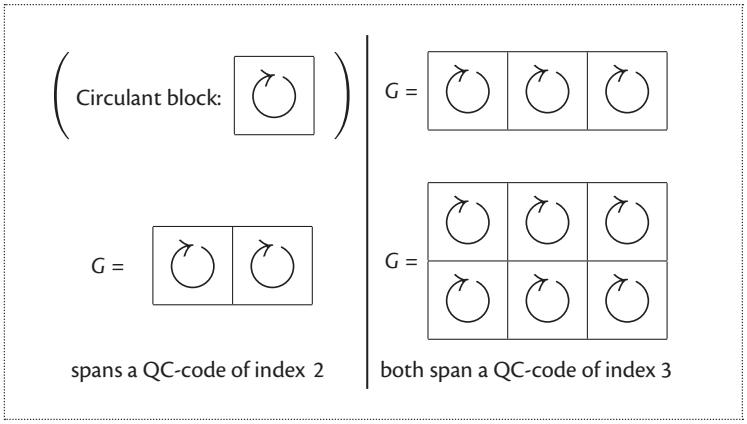

Figure 3. Block-circulant matrices.

Representing a block-circulant matrix requires only the first row of each circulant block. When the index is small, this leads to considerably smaller public keys.

Circulant binary matrices of size $p \times p$ form a commutative ring isomorphic to the quotient ring of polynomial $\mathbf{F}_2[x]/(x^p - 1)$. To a circulant matrix whose first row is $(a_0, a_1, \ldots, a_{p-1})$, I associate the polynomial $a(x) = a_0 + a_1 x + \cdots + a_{p-1}x^{p-1}$. Matrix addition, multiplication, and inversion correspond to the same polynomial operations modulo $x^p - 1$. The Hamming weight of $a(x)$ is the Hamming weight of the binary word $(a_0, a_1, \ldots, a_{p-1})$.

Algebraic Quasicyclic Codes

Algebraic codes contain algebraic structure allowing mathematical description and fast decoding algorithms. The coding theory domain is vast, but for cryptographic purposes, we can restrict it to the subclass of alternant codes, among which are the binary Goppa codes used by McEliece. If a McEliece-type cryptosystem uses a family of alternant codes, it's possible to derive from the public key an overdetermined system of multivariate polynomial equations over a finite field. If the hidden code is alternant, this polynomial system has a solution, but finding it is intractable in general.

If the code rate is close to 1, the system is highly overdetermined, and the subsystem obtained by considering only low degree equations—though it can't be solved—has particular properties leading to a distinguisher.[6]

If the code is quasicyclic, the number of unknowns decreases, particularly when the index is small. Solving the system might become tractable as demonstrated in Jean-Charles Faugère and his colleagues'

Parameters: block size p, row weight w, error weight t, $\mathcal{R}_p = \mathbf{F}_2[x]/(x^p - 1)$
(p a prime, w even, $w/2$ odd, w and t are close to $\sqrt{2p}$)
Key generation: pick h_0 and h_1 in $\mathcal{R}_p$ both of weight $w/2$
public key: $g = h_1 h_0^{-1}$
private key: h_0, h_1
Encryption: $\mathcal{R}_p \rightarrow \mathcal{R}_p \times \mathcal{R}_p$
$m \mapsto (mg + e_0, mg + e_1)$ with $wt(e_0) + wt(e_1) = t$
Decryption: given a ciphertext (u_0, u_1)
solve $u_0 h_0 + u_1 h_1 = e_0 h_0 + e_1 h_1$ with $wt(e_0) + wt(e_1) \leq t$

Figure 4. Quasicyclic moderate density parity check (QC-MDPC)-McEliece scheme.

"Algebraic Cryptanalysis of McEliece Variants with Compact Keys."[14]

To say it crudely, the algebraic and the quasicyclic structures combine to provide a simpler attack. Note that simpler doesn't mean feasible, and not all compact-key variants of McEliece based on alternant codes have been broken so far. Nevertheless, pursuing this line of work requires a full understanding of how far the polynomial system solvers can go to solve the above-mentioned systems.

Quasicyclic MDPC Codes

Another approach, proposed by Rafael Misoczki and his colleagues in "MDPC-McEliece: New McEliece Variants from Moderate Density Parity-Check Codes,"[15] was to consider MDPC codes. An MDPC code possesses a moderately sparse parity check matrix, meaning that the rows have length n and Hamming weight of order $\sqrt{n}$. Knowledge of the sparse parity check matrix allows the decoding of an error of weight proportional to $\sqrt{n}$.

The public key is any generator matrix of that code. Such a matrix is dense in general and doesn't reveal anything about the secret sparse parity check matrix. The sparse parity check matrix can be block circulant, in which case the public generator matrix is also block circulant and the corresponding MDPC code is quasicyclic. The whole scheme can then be described in terms of polynomials in the quotient ring $\mathcal{R}_p = \mathbf{F}_2[x]/(x^p - 1)$. Figure 4 shows the scheme for quasicyclic MDPC (QC-MDPC) codes of index 2 and rate $k/n = 1/2$.

Parameter selection. For 128 bits of security, with QC-MDPC codes of index 2, the parameter sizes are $p = 9{,}857$, $w = 142$, and $t = 134$, leading to a public key of size 9,857 bits (about 1.2 Kbytes). Against quantum adversaries, the public key size must increase to 32,771 bits, about 4 Kbytes, instead of 1 Mbyte for the original Goppa-McEliece.

Decoding. Decoding QC-MDPC codes consists of solving the following problem.

Problem 1 (QC-MDPC decoding): Given s, h_0, h_1 in $\mathcal{R}_p$ with $wt(h_0) = wt(h_1) = w/2$, find e_0, e_1 in $\mathcal{R}_p$ such that $wt(e_0) + wt(e_1) \leq t$ and $e_0 h_0 + e_i h_j = s$.

The interested reader can easily check that whoever can solve that problem can perform the decryption in Figure 4. Problem 1 is easy as long as w and t are both small compared to the code length. In fact, the product of t by w must be of the same magnitude as the code length $n = 2p$. Various known algorithms can solve that problem; the exact decoding performance will depend on the chosen algorithm and the parameters must be validated using simulations or analysis.

Security proof. QC-MDPC-McEliece security is provably reduced to the two following problems related to sparse binary polynomials in the ring $\mathcal{R}_p = \mathbf{F}_2[x]/(x^p - 1)$.[4]

Problem 2 (QC decoding): Given s, g in $\mathcal{R}_p$, find e_0, e_1 in $\mathcal{R}_p$ such that $wt(e_0) + wt(e_1) \leq t$ and $e_0 + e_1 g = s$.

Problem 3 (QC-MDPC distinguishing): Given g in $\mathcal{R}_p$, is there h_0, h_1 in $\mathcal{R}_p$ such that $wt(h_0) + wt(h_1) \leq w$ and $h_1 + h_0 g = 0$?

Problem 2 is the generic decoding problem restricted to quasicyclic codes. Problem 3 entails deciding whether a given quasicyclic code contains a word of small weight. Those problems are exactly the quasicyclic counterparts of the two NP-complete problems stated by Elwyn Berlekamp and his colleagues in "On the Inherent Intractability of Certain Coding Problems."[2] They are believed to be hard on average, even though, formally, the arguments Michael Alekhnovich gives in "More on Average Case vs Approximation Complexity" don't hold for the noncyclic case.[5]

Key exchange protocol. Because key generation is very easy, QC-MDPC-McEliece is very well suited for exchanging session keys. Figure 5 shows a sketch of a key exchange protocol. At the end of the protocol, the two parties involved, here Alice and Bob, share a secret K, valid for one session, to typically be used as the key of a symmetric encryption scheme like the Advanced Encryption Standard. Because the public key g is used only once, this protocol achieves forward secrecy, an extremely desirable feature that protects past sessions against future compromises of secret keys.

This key exchange protocol is particularly simple, and its security, as for the encryption scheme, is provably reduced to the hardness of Problems 2 and 3.

Decoding failures. Finally, the current choice of parameters in Misoczki and his colleagues' "MDPC-McEliece: New McEliece Variants from Moderate Density Parity-Check Codes"[15] leads to a small probability of failure with state-of-the-art MDPC decoders. This

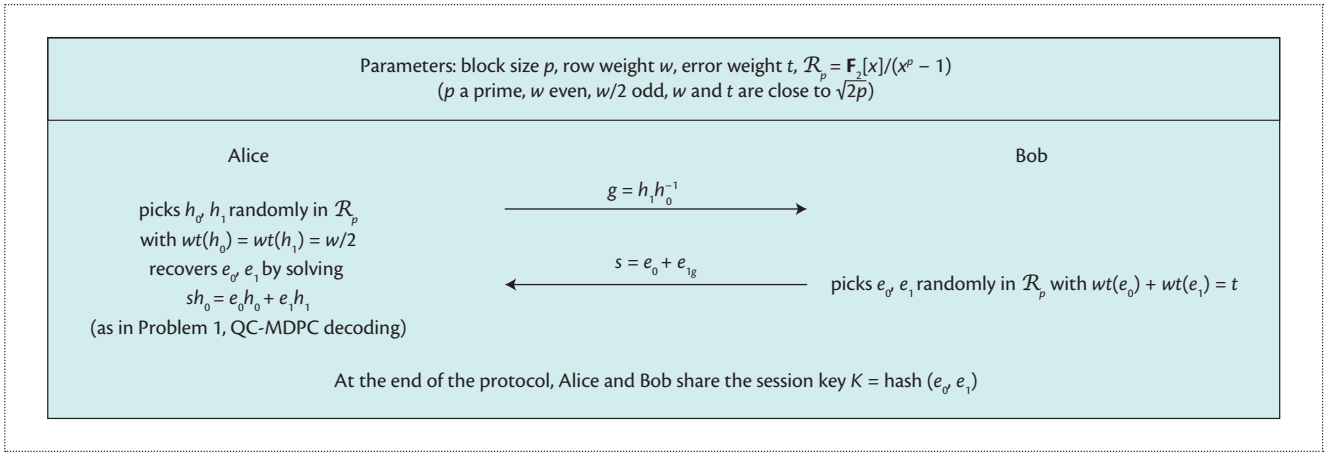

Figure 5. QC-MDPC key exchange protocol. The two parties involved, Alice and Bob, share a secret *K*, valid for one session, to be used as the key of a symmetric encryption scheme like the Advanced Encryption Standard.

probability is low enough be a benign nuisance in normal usage of the scheme. However, very recently Qian Guo and his colleagues demonstrated that error patterns leading to decoding failures are correlated with the secret key.[16] If adversaries have the ability to collect many error patterns leading an MDPC decoding failure, they can recover the secret. We can counter this attack by reducing the probability of a decoding failure to a small enough quantity. This can be achieved by increasing the block size (and thus the key size). The precise amount by which to increase the block size is an open research question.

Finally, note that the forward secrecy key exchange protocol presented here isn't vulnerable to this attack because each public key is used only once. In the case of a decoding failure, the protocol restarts from the beginning with a fresh public key *g*.

After decades of research, code-based cryptography has reached a certain maturity. The original McEliece encryption scheme is a very strong candidate as one of the future quantum-resistant standards for public-key encryption that must be defined in the coming decade. Its main limitation is a relatively large key size (on the order of 1 Mbyte for the quantum-resistant variant with long-term security), which makes it less suitable for some applications.

The other interesting candidates for code-based quantum-resistant cryptography are the variants based on QC-MDPC codes, which allow for much shorter keys and could provide a simple and efficient key exchange protocol with forward secrecy. Moreover, its security provably reduces to the hardness of two well-identified coding theory problems, both of which are believed to be hard.

Code-based cryptography has a thorn on its side with the absence of a satisfying digital-signature scheme. The most credible possibility at this time would be a non-interactive variant of Jacques Stern's zero-knowledge authentication scheme,[8] but this would mean relatively large signature sizes (one or a few hundred kilobytes). Finding other, more convenient digital signature primitives is certainly one of the major challenges in code-based cryptography today. ■

References

1. R.J. McEliece, *A Public-Key Cryptosystem Based on Algebraic Coding Theory*, Deep Space Network progress report, Jet Propulsion Lab., California Inst. Technology, Jan. 1978, pp. 114–116.
2. E.R. Berlekamp et al., "On the Inherent Intractability of Certain Coding Problems," *IEEE Trans. Information Theory*, vol. 24, no. 3, 1978, pp. 384–386.
3. N. Courtois, M. Finiasz, and N. Sendrier, "How to Achieve a McEliece-Based Digital Signature Scheme," *Advances in Cryptology* (ASIACRYPT 01), LNCS 2248, Springer, 2001, pp. 157–174.
4. N. Sendrier, "On the Use of Structured Codes in Code Based Cryptography," *Coding Theory and Cryptography III*, S. Nikova, B. Preneel, and L. Storme, eds., 2009, pp. 59–68.
5. M. Alekhnovich, "More on Average Case vs Approximation Complexity," *Proc. 44th Symp. Foundations of Computer Science* (FOCS 03), 2003, pp. 298–307.
6. J.-C. Faugère et al., "A Distinguisher for High-Rate McEliece Cryptosystems," *IEEE Information Theory Workshop* (ITW 11), 2011, pp. 282–286.
7. H. Niederreiter, "Knapsack-Type Cryptosystems and Algebraic Coding Theory," *Problems of Control and Information Theory*, vol. 15, no. 2, 1986, pp. 157–166.
8. J. Stern, "A New Identification Scheme Based on

Syndrome Decoding," *Advances in Cryptology* (CRYPTO 93), LNCS 773, Springer, 1993, pp. 13–21.
9. A. Fiat and A. Shamir, "How to Prove Yourself: Practical Solutions to Identification and Signature Problems," *Advances in Cryptography* (CRYPTO 86), LNCS 263, Springer, 1982, pp. 186–194.
10. D. Unruh, "Non-interactive Zero-Knowledge Proofs in the Quantum Random Oracle Model," *Advances in Cryptography* (EUROCRYPT 15), LNCS 9057, Springer, 2015, pp. 755–784.
11. K. Kobara and H. Imai, "Semantically Secure McEliece Public-Key Cryptosystems—Conversions for McEliece PKC," *Public Key Cryptography*, LNCS 1992, Springer, 2001, pp. 19–35.
12. D.J. Bernstein et al., "McBits: Fast Constant-Time Code-Based Cryptography," *Proc. Int'l Workshop Cryptographic Hardware and Embedded Systems* (CHES 13), LNCS 8086, Springer, 2013, pp. 250–272.
13. P. Gaborit, "Shorter Keys for Code Based Cryptography," *Proc. Int'l Workshop Coding and Cryptography* (WCC 05), 2005, pp. 81–90.
14. J.-C. Faugère et al., "Algebraic Cryptanalysis of McEliece Variants with Compact Keys," *Advances in Cryptology* (EUROCRYPT 10), LNCS 6110, Springer, 2010, pp. 279–298.
15. R. Misoczki et al., "MDPC-McEliece: New McEliece Variants from Moderate Density Parity-Check Codes," *Proc. IEEE Int'l Symp. Information Theory* (ISIT 13), 2013, pp. 2069–2073.
16. Q. Guo, T. Johansson, and P. Stankovski, "A Key Recovery Attack on MDPC with CCA Security Using Decoding Errors," *Advances in Cryptology* (ASIACRYPT 16), LNCS 10031, Springer, 2016, pp. 789–815.

Nicolas Sendrier is a senior research scientist at INRIA. His main research interests include the design and analysis of code-based cryptographic primitives. Sendrier received a PhD in computer science from University Paris 6. He's a steering committee member of the Postquantum Cryptography conference series. Contact him at nicolas.sendrier@inria.fr.