CAKE: Code-based Algorithm for Key Encapsulation

Paulo S. L. M. Barreto[1,2], Shay Gueron[3,4], Tim Güneysu[5,6], Rafael Misoczki[7], Edoardo Persichetti[8], Nicolas Sendrier[9], Jean-Pierre Tillich[9]

[1] University of Washington Tacoma, USA
[2] University of São Paulo, Brazil
[3] University of Haifa, Israel
[4] Amazon Web Services**, USA
[5] University of Bremen, Germany
[6] DFKI, Germany
[7] Intel Corporation, USA
[8] Florida Atlantic University, USA
[9] INRIA, France

Abstract. Current widely-used key-exchange (KE) mechanisms will be vulnerable to quantum attacks when sufficiently strong quantum computers become available. Therefore, devising quantum-resistant replacements that combine efficiency with solid security guarantees is an important and challenging task. This paper proposes several contributions towards this goal. First, we introduce "*CAKE*", a key-encapsulation algorithm based on the QC-MDPC McEliece encryption scheme, with two major improvements: a) the use of ephemeral keys that defeats a recent reaction-attack against MDPC decoding of the corresponding encryption scheme and b) a highly efficient key-generation procedure for QC-MDPC-based cryptosystems. Then, we present an authenticated key-exchange protocol based on CAKE, which is suitable for the Internet Key-Exchange (IKE) standard. We prove that CAKE is IND-CCA secure, that the protocol is SK-Secure, and suggest practical parameters. Compared to other post-quantum schemes, we believe that CAKE is a promising candidate for post-quantum key-exchange standardization.

Keywords: post-quantum cryptography, code-based cryptography, key exchange.

** This work was done prior to joining Amazon.

Acknowledgments: Shay Gueron, Tim Güneysu, Nicolas Sendrier and Jean-Pierre Tillich were supported in part by the Commission of the European Communities through the Horizon 2020 program under project number 645622 (PQCRYPTO). Shay Gueron was also partially supported by the Israel Science Foundation (grant No. 1018/16). Paulo S. L. M. Barreto was partially supported by Intel and FAPESP through the project "Efficient Post-Quantum Cryptography for Building Advanced Security Applications" (grant No. 2015/50520-6).

1 Introduction

The currently deployed public-key cryptosystems rely on the difficulty of number theory problems (e.g., factorization), and discrete logarithm problems [38, 31]. These problems will be efficiently solved by large quantum computers [41], thus turning those schemes completely useless in a not-so-distant future. Therefore, it is of great relevance to devise and deploy alternative schemes that can survive the advent of large quantum computers and, ideally, still offer reasonable performance.

In this context code-based cryptography is a promising alternative. It relies on the well-known decoding problem [3], believed to be hard even against quantum adversaries [4], and on the indistinguishability of its public key from random, a problem that strongly depends on the code family. The best-known code-based scheme, namely the McEliece encryption scheme [29], suggests binary Goppa codes as the code family. However, this choice has two main drawbacks: a) Goppa codes may not be the optimal security choice given a recent distinguisher for certain Goppa codes [13] and b) they require very large public keys (of several megabytes size).

To address these issues, the QC-MDPC McEliece scheme [32] was introduced replacing Goppa codes by Quasi-Cyclic Moderate-Density Parity-Check (QC-MDPC) codes. This approach led to key sizes that are comparable to RSA keys, and approximated the distinguishing problem to the decoding problem. These features attracted great attention from the community (see [20, 43, 44, 11], just to mention a few) including a mention in the preliminary European recommendations for post-quantum cryptography [25].

Despite their promising features, QC-MDPC codes need to be handled carefully due to the probabilistic nature of MDPC decoding (which is inherited from Low-Density Parity-Check (LDPC) codes [16]): there is some probability that the MDPC decoding may fail. This property can be leveraged to mount an attack on some schemes. Indeed, Guo, Johansson and Stankovski [17] presented an interesting reaction attack (GJS attack) against the QC-MDPC McEliece encryption scheme. In this attack, the adversary carefully crafts error patterns and observes whether (or not) the decoding process fails. The adversary can recover the private key by collecting the decoding failure rate of various error patterns. This is possible only because the parameters suggested in [32] ensured (through exhaustive simulation) a decoding failure rate (DFR) of 10^{-7}. The simplest way to foil this attack is to choose parameters such that DFR $\leq 2^{-\lambda}$, where λ is the security level of the scheme. However, the difficulty with this strategy is to *formally* prove that a given parameter set attains a given DFR. Empirical observations indicated that the DFR of MDPC codes decreases exponentially, and secure parameters could be achieved by increasing the code length of [32] by 30% or 40%. However, with no formal proof for this property, the GJS attack might prevent wide adoption of QC-MDPC McEliece for asymmetric encryption.

Contributions. To date neither an MDPC-based key exchange protocol nor any MDPC-based encryption scheme has been proposed that defeats the GJS attack. This paper provides several contributions that address this and other problems:

- ✓ It introduces CAKE, a new key encapsulation mechanism (KEM), based on QC-MDPC codes. It differs from the QC-MDPC McEliece encryption scheme in two respects: a) the key generation process is significantly faster at the cost of longer public keys, and b) completely defeats the GJS attack by employing ephemeral keys (i.e., new keys are generated at each key exchange).
- ✓ It proposes an authenticated key exchange protocol based on CAKE that is suitable for the Internet Key Exchange (IKE), similarly to what was done done for lattices [33].
- ✓ It proves that CAKE is CCA secure and the protocol is SK secure [9].

The full version of this paper will include a discussion of implementation aspects, including strategies to deploy our proposal in an isochronous way.

Related Work. Lattice-based cryptography has a long record of academic works [34], including promising KE protocols. The NewHope scheme [1] provides good performance and is based on the Ring-LWE problem [26] (a ring variant of the Learning-With-Errors (LWE) problem [37]). It improves over previous work by Bos, Costello, Naherig and Stebilla [7] which is an implementation of Peikert's proposal [33] for TLS. Frodo [6] is a key exchange scheme based on the LWE problem itself at the price of larger parameters and lower performance. Cryptography based on isogenies of supersingular elliptic curves seems to be another promising way to devise KE protocols [22, **?**] offering small public-keys but not so attrative latency. The only known code-based key encapsulation mechanism (KEM) scheme is McBits [5]. It builds on the work of [35], and lives in the classical McEliece setting with binary Goppa codes and enormous public keys.

Organization. This paper is organized as follows. Section 2 presents the preliminary concepts, Section 3 introduces CAKE, a new unauthenticated key encapsulation mechanism (KEM) based on QC-MDPC codes, Section 4 presents an authenticated key exchange protocol based on CAKE, Section 5 proves that CAKE is IND-CCA secure and the corresponding authenticated key exchange protocol is SK secure, Section 6 discusses practical security and suggests parameters. Section 7 presents our conclusions.

2 Preliminaries

Definition 1 (Linear codes). *The Hamming weight of a vector $x \in \mathbb{F}_2^n$ is the number $\mathsf{wt}(x)$ of its nonzero components. A* binary (n, r)-linear code $\mathcal{C}$ *of length n, co-dimension r and dimension $k = (n - r)$ is a k-dimensional vector subspace of $\mathbb{F}_2^n$. It is spanned by the rows of a matrix $G \in \mathbb{F}_2^{k \times n}$, called a* generator matrix *of $\mathcal{C}$. Equivalently, it is is the kernel of a matrix $H \in \mathbb{F}_2^{r \times n}$, called* parity-check matrix*, i.e. $\mathcal{C} = \{c \mid Hc^T = 0\}$. The* codeword $c \in \mathcal{C}$ *of a vector $m \in \mathbb{F}_2^{(n-r)}$ is $c = mG$. The* syndrome $s \in \mathbb{F}_2^r$ *of a vector $e \in \mathbb{F}_2^n$ is $s^T = He^T$.*

Definition 2 (Quasi-cyclic code). *An (n, r)-linear code is quasi-cyclic (QC) if there is some integer n_0 such that every cyclic shift of a codeword by n_0 places is again a codeword.*

When $n = n_0 r$, for some integer r, it is possible and convenient to have both generator and parity check matrices composed by $r \times r$ circulant blocks. A circulant block is completely described by its first row (or column) and the algebra of $r \times r$ binary circulant matrices is isomorphic to the algebra of polynomials modulo $x^r - 1$ over $\mathbb{F}_2$, enabling efficient computations. For example, a parity-check matrix H of an $(n_0 r, r)$-quasi-cyclic code can be represented as:

$$H = [H_0 | \dots | H_{n_0-1}], \text{ where: } H_i = \begin{pmatrix} h_{i,0} \dots h_{i,r-1} \\ \vdots \quad \ddots \quad \vdots \\ h_{i,1} \dots \quad h_{i,0} \end{pmatrix} \in \mathbb{F}_2^{r \times r}$$

Definition 3 (QC-MDPC codes). *An (n_0, r, w)-QC-MDPC code is a quasi-cyclic code of length $n = n_0 r$, codimension r admitting a parity-check matrix with constant row weight $w = O(\sqrt{n \log n})$.*

3 CAKE: A QC-MDPC KEM with Fast Key-Generation

In this section we introduce CAKE – an unauthenticated key-encapsulation mechanism based on QC-MDPC codes. The strategy to present our scheme as an unauthenticated KEM follows works such as NewHope [6] and BCNS [7]. In this way, authentication and key exchange features are decoupled, allowing flexibility to select (and eventually replace) the choice for each feature. Section 4 describes one way to add the authentication layer on top of CAKE.

CAKE resembles the QC-MDPC McEliece encryption scheme [32] but also has important differences. While QC-MDPC McEliece intends to use long term keys, CAKE relies on ephemeral keys. This means that a new key pair is generated at each key exchange, thus completely defeating the GJS attack [17] which depends on observing a large number of decoding failures for a same private key. Given the new requirement of generating a key pair at every key exchange, a major challenge consisted of investigating novel strategies to accelerate MDPC key generation. After several attempts, we came up with a simple and elegant solution. In contrast to QC-DMPC McEliece (and any quasi-cyclic McEliece variant), CAKE does not compute a polynomial inversion in the cyclic ring $(x^r - 1)$, then multiply it by the sparse private matrix to hide the private code structure. Instead, CAKE hides the private code structure by simply multiplying it by any random, dense polynomial. This turns CAKE key generation as efficient as QC-MDPC encryption. The drawback of this strategy is the doubled size of the public key since the public key will not have an identity block; an acceptable cost given the significant speedup. Finally, we make use of a simple variant of McEliece, as presented in [10]. We swap the roles of message and randomness in the encryption process to avoid a costly polynomial inversion. A detailed description of this message-randomness tweak is provided in Appendix B.

A key encapsulation mechanism (KEM) is composed by three algorithms: Gen which outputs a public encapsulation key pk and a private decapsulation key sk, Encaps which takes as input an encapsulation key pk and outputs a ciphertext C and a symmetric key K, and Decaps which takes as input a decapsulation key sk and a ciphertext C and outputs a symmetric key K or a decapsulation failure symbol $\perp$. For more details on KEM definitions, we refer the reader to [12]. The standard security definitions for a KEM are given in Appendix C.

For a target security level λ, let r be a prime such that $(x^r-1)/(x-1) \in \mathbb{F}_2[x]$ is irreducible, let d_v be an odd integer and let t be an integer such that decoding t errors with a uniformly chosen binary linear error-correcting code of length $n = 2r$ and dimension r, as well as recovering a base of column weight d_v given an arbitrary base of a QC-MDPC code of the same length and dimension, both have a computational cost in $\Omega(\exp(\lambda))$. See Section 6 for a detailed discussion on how to securely select such parameters. We now define CAKE as follows:

Algorithm 1. CAKE.Gen:

- Input: λ, the target quantum security level.
- Output: the sparse private key (h_0, h_1) and the dense public key (g_0, g_1).

0. Given λ, set the parameters r, d_v, t as described above.
1. Choose $h_0, h_1 \xleftarrow{\$} \mathbb{F}_2[x]/\langle x^r - 1\rangle$ both of (odd) weight d_v.
2. Choose $g \xleftarrow{\$} \mathbb{F}_2[x]/\langle x^r - 1\rangle$ of odd weight (so $\mathsf{wt}(g) \approx r/2$).
3. Compute $(g_0, g_1) \leftarrow (g \cdot h_1^T, g \cdot h_0^T)$.

Let H and G be the quasi-cyclic matrices built from $(r-1)$ cyclic shifts of (h_0, h_1) and (g_0, g_1) respectively. It is easy to see that $G \cdot H^T = 0$ and therefore they satisfy the condition to be a generator and a parity-check matrix of the given code: $G \cdot H^T = [g \cdot h_1^T \mid g \cdot h_0^T] \cdot [h_0 \mid h_1]^T = g \cdot h_1^T \cdot h_0^T + g \cdot h_0^T \cdot h_1^T = g \cdot (h_1^T \cdot h_0^T + h_0^T \cdot h_1^T) = 2 \cdot g \cdot h_0^T \cdot h_1^T = 0$. It is also important to show that g, as created above, is always invertible (thus not risking to generate a public-code which is in fact a sub-code of the private one) and this is proven in Appendix A.

The encapsulation and decapsulation algorithms make use of three hash functions $\mathbf{G} : \{0,1\}^n \rightarrow \{0,1\}^r$, $\mathbf{C} : \{0,1\}^* \rightarrow \{0,1\}^{\ell_H}$ and $\mathbf{H} : \{0,1\}^{\ell_H} \rightarrow \{0,1\}^{\ell_H}$, where ℓ_H is the digest length of a hash function that offers λ bits of collision-resistance security against a quantum adversary. The first has the task of generating randomness for the scheme, the second compresses the sparse error vector into a digest of ℓ_H bits and the third is a length-preserving hash that provides "plaintext confirmation" as in [21]. The shared symmetric key is obtained via another hash function $\mathbf{K} : \{0,1\}^{2n} \rightarrow \{0,1\}^{\ell_K}$, where ℓ_K is the desired key length. Public and private key are n bits long and the cryptogram is $(n + \ell_H)$ bits long.

Algorithm 2. CAKE.Encaps:

- Input: the dense public key (g_0, g_1).
- Output: the encapsulated key K and the cryptogram $C = (c, d)$.

1. Generate an error pattern $e_0, e_1 \xleftarrow{\$} \mathbb{F}_2[x]/\langle x^r - 1\rangle$ of total weight t.
2. Set $e = (e_0, e_1)$, then compute $m = \mathbf{G}(e)$, $d_0 = \mathbf{C}(e)$ and $d = \mathbf{H}(d_0)$.
3. Compute $c = (c_0, c_1) \leftarrow (m \cdot g_0 + e_0, m \cdot g_1 + e_1)$.
4. Compute $K \leftarrow \mathbf{K}(c, e)$ and set $C = (c, d)$.

Algorithm 3. CAKE.Decaps:

- Input: the sparse private key (h_0, h_1) and the cryptogram C.
- Output: the decapsulated key K or a failure symbol $\perp$.

1. Compute the syndrome $s \leftarrow c_0 \cdot h_0^T + c_1 \cdot h_1^T$.
2. Try to decode s to recover an error vector $e' = (e'_0, e'_1)$.
3. If $\mathsf{wt}(e') \neq t$ or decoding fails, output $\perp$ and halt.
4. Compute $m' = \mathbf{G}(e')$, $d'_0 = \mathbf{C}(e')$ and $d' = \mathbf{H}(d'_0)$.
5. Compute $c' \leftarrow (m' \cdot g_0 + e'_0, m' \cdot g_1 + e'_1)$.
6. If $c' \neq c \vee d' \neq d$, output $\perp$ and halt.
7. Compute $K \leftarrow \mathbf{K}(c, e')$.

Figure 1 illustrates CAKE as a protocol of messages exchanged between an Initiator, who starts the key exchange process, and a Responder.

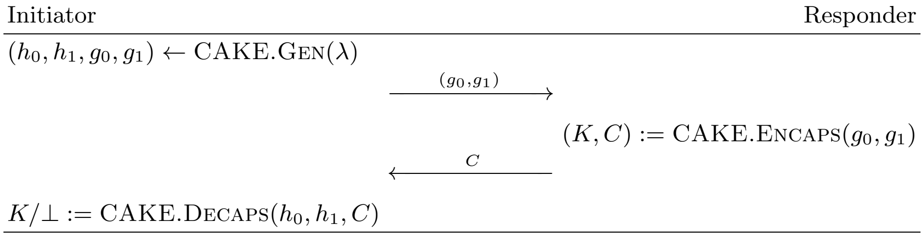
Initiator
Responder
$(h_0, h_1, g_0, g_1) \leftarrow$ CAKE.Gen(λ)
(g_0, g_1)
$(K, C) :=$ CAKE.Encaps(g_0, g_1)
C
$K/\perp :=$ CAKE.Decaps(h_0, h_1, C)

Fig. 1: CAKE Key Encapsulation Mechanism

4 An Authenticated Key Exchange Protocol From CAKE

In this section, we discuss one way to extend CAKE to an authenticated key exchange protocol. This discussion intends to demonstrate that CAKE ephemeral keys are not a limitation for its integration into real-world key exchange protocols and also allows us to discuss interesting security properties required in the real world, such as perfect forward secrecy, which are usually managed in levels of abstraction above the simple key encapsulation building-block.

The construction here described is based on the well-known SIGn-and-MAc (SIGMA) protocol design [24], which is adopted by the Internet Key Exchange (IKE) protocol [19], part of the IPSec standard [40]. The simplest SIGMA protocol is known as Σ_0 and is proven to be secure (in terms to be discussed in Section 5.2) when instantiated with Diffie-Hellman key agreement [9]. Our proposal essentially leverages a result presented by Peikert which demonstrated that Σ_0 can be proven secure with any IND-CPA KEM [33], instead of being restricted to Diffie-Hellman.

As in [33], the key exchange protocol here described is parametrized by an (IND-CPA-secure) key encapsulation mechanism KEM with key space K, a digital signature scheme SIG, a pseudorandom function $f : \mathsf{K} \times \{0,1\} \rightarrow \mathsf{K}_0$, and a message authentication code MAC with key space K_0 and message space $\{0,1\}^*$. A successful execution of the protocol outputs a secret key in K_0. In our work, we explicitly defines CAKE as the KEM scheme. For the sake of flexibility, we do not specify any particular signature, MAC or pseudorandom function although they all need to meet some minimum security notion (the signature and MAC must be EUF-CMA secure and f must be a secure pseudorandom function; see Section 5). We assume that each party has a long-term signing key for SIG whose corresponding verification key is publicly available and associated to its identity ID. This can be done in terms of certificate authorities and common public-key infrastructure.

Key exchange protocols are multiparty protocols activated by messages that are locally processed, leading to new messages being triggered. A session is an invocation of this protocol. Each session is associated to a unique session ID (denoted as sid) and a party can be called the Initiator (with identity ID_I) who first activates the session or the Responder (identity ID_R) who is activated upon receiving a message. For a more detailed discussion on key exchange protocol definitions we refer to [8]. Figure 2 describes how CAKE can be plugged into an authenticated key exchange protocol, similarly as done in [33].

The protocol assumes that Initiator and Responder possess identities ID_I and ID_R, respectively. Initiator generates a unique session identifier sid and a CAKE key pair $(sk = (h_0, h_1), pk = (g_0, g_1))$, and sends (sid, pk) to Responder, who generates a key K and a ciphertext C using the encapsulation method. The pair (K_0, K_1) is generated from K using the pseudorandom function f. The tuple $(1, sid, pk, C)$ is signed using Responder's signing key and a MAC tag is generated from $(1, dif, ID_R)$ using key K_1. The signature, tag, ID_R, sid and C is sent to Initiator, who tries to decapsulate C. In case of success, Initiator reconstructs (K_0, K_1) and verifies both signature and MAC tag. If it succeeds, Initiator signs the tuple $(0, sid, C, pk)$ and generates a MAC tag for the tuple $(0, sid, ID_I)$. Signature, tag, sid and ID_I are sent to Responder who verifies both signature and tag. If it succeeds, the public output is the tuple (ID_I, sid, ID_R) and the local output is the shared key K_0. If any process fails, the public output is $(abort, ID_I, sid)$ and $(abort, ID_R, sid)$, and the key exchange is restarted.

Initiator		Responder
ID_I		ID_R
sid		
(sk, pk) :=CAKE.Gen(λ)		
	$\xrightarrow{sid,pk}$	
		(K, C) := CAKE.Encaps(pk)
		$K_0 := f_K(0),\ K_1 := f_K(1)$
		$\alpha := SIG.Sign_R(1, sid, pk, C)$
		$\beta := MAC.Tag_{K_1}(1, sid, ID_R)$
	$\xleftarrow{sid,C,ID_R,\alpha,\beta}$	
k or $\perp$:= CAKE.Decaps(sk, C)		
$K_0 := f_K(0),\ K_1 := f_K(1)$		
$\top$ or $\perp := SIG.Verify_R(\alpha)$		
$\top$ or $\perp := MAC.Verify_{K_1}(\beta)$		
$\gamma := SIG.Sign_I(0, sid, C, pk)$		
$\delta := MAC.Tag_{K_1}(0, sid, ID_I)$		
	$\xrightarrow{sid,ID_I,\gamma,\delta}$	
		$\top$ or $\perp := SIG.Verify_I(\gamma)$
		$\top$ or $\perp := MAC.Verify_{K_1}(\delta)$
Public output: (ID_I, sid, ID_R)		Public output: (ID_I, sid, ID_R)
Local output: (K_0)		Local output: (K_0)

Fig. 2: SIGMA-like Authenticated Key Exchange From CAKE KEM

5 Formal Security Assessment

In this section, we prove that CAKE is IND-CCA secure and that the authenticated key-exchange protocol described in Section 4 is SK-Secure.

5.1 CAKE IND-CCA Security

We will provide two main security results regarding CAKE, but we first introduce a definition regarding generic encryption schemes.

Definition 4. *Consider a probabilistic PKE with randomness set* R. *We say that PKE is* γ-spread *if for a given key pair* (sk, pk)*, a plaintext* m *and a string* y *in the ciphertext domain, we have*

$$\mathsf{Pr}[u \xleftarrow{\$} \mathsf{R} \mid y = \textsc{Enc}_{pk}(m, u)] \le 2^{-\gamma},$$

for a certain $\gamma \in \mathbb{R}$.

The definition above is presented as in [21], but note that in fact this corresponds to the notion of γ-*uniformity* given by Fujisaki and Okamoto in [15], except for a change of constants. In other words, a scheme is γ-spread if it is $2^{-\gamma}$-uniform.

Proposition 1. *The KEM scheme CAKE has the same correctness error ϵ as the underlying PKE. Moreover, let PKE be γ-spread. Then for any IND-CCA adversary $\mathcal{A}$ against CAKE that makes at most $q_{\mathbf{G}}$ many queries to the random oracle $\mathbf{G}$, at most $q_{\mathbf{K}}$ many queries to the random oracle $\mathbf{K}$, and q_D queries to the decryption oracle, there exists a OW-CPA adversary $\mathcal{B}$ against PKE, running in approximately the same time as $\mathcal{A}$, such that*

$$\mathsf{Adv}_{KEM}^{IND-CCA}(\mathcal{A}) \leq q_{\mathbf{K}} \cdot (\epsilon + 2^{-\gamma}) + (q_{\mathbf{G}} + 1) \cdot \mathsf{Adv}_{PKE}^{OW-CPA}(\mathcal{B}).$$

Proof. The theorem follows quite straightforwardly from [21, Cor. 3.6]. Namely, our protocol fits into the $\mathsf{FO}^{\perp}$ framework for transforming a public-key encryption scheme into a KEM. The proof uses standard techniques (i.e. a sequence of games) and shows that breaking IND-CCA security of the KEM would lead to break the OW-CPA security of the underlying encryption scheme. □

Note that the value d included in the KEM cryptogram is not necessary for the security result above, but it is a crucial factor to provide security in the Quantum Random Oracle Model (QROM), given by the next proposition.

Proposition 2. *The KEM scheme CAKE has the same correctness error ϵ as the underlying PKE. Moreover, for any quantum IND-CCA adversary $\mathcal{A}$ against CAKE that makes at most $q_{\mathbf{G}}, q_{\mathbf{H}}, q_{\mathbf{K}}$ many queries to, respectively, random oracles $\mathbf{G}, \mathbf{H}$ and $\mathbf{K}$, and q_D (classical) queries to the decryption oracle, there exists a quantum OW-CPA adversary $\mathcal{B}$ against PKE, running in approximately the same time as $\mathcal{A}$, such that*

$$\mathsf{Adv}_{KEM}^{IND-CCA}(\mathcal{A}) \leq 4(q_{\mathbf{K}} + q_{\mathbf{H}}) \cdot \sqrt{q_D q_{\mathbf{H}} \cdot \epsilon + q_{\mathbf{G}} \cdot \sqrt{\mathsf{Adv}_{PKE}^{OW-CPA}(\mathcal{B})}}.$$

Proof. The theorem follows quite straightforwardly from [21, Cor. 4.4]. Namely, our protocol fits into the $\mathsf{QFO}^{\perp}$ framework for transforming a public-key encryption scheme into a KEM. Again, the proof uses a sequence of games, but in the quantum random oracle model (i.e. hash functions are modeled as quantum random oracles) and shows that breaking IND-CCA security of the KEM would lead to break the OW-CPA security of the underlying encryption scheme. □

5.2 SK Security of Authenticated Key Exchange from CAKE

The security notion targeted by our SIGMA-like construction and also by [33] is known as SK Secure[1], which stands for *session-key secure* [9]. Informally, this notion translates into: "the adversary does not learn anything about the session key by interacting with the protocol" and enables the establishment of secure channels (usually the ultimate goal of sharing a key). In the following paragraphs, we give an overview on SK Security.

According to [9], a key exchange protocol is a multiparty protocol where each party runs one or more copies of the protocol. A *session* is a local procedure resulting from a protocol activation at a party. The activation of a protocol at a party has three inputs (P, sid, d): the local party P, the unique session identifier sid and the intended peer address d. A party can be activated as an Initiator or as a Responder (upon an incoming message). The output of a session is a public triple (P, sid, Q), where Q is the intended peer identity and a secret *session key.* In case of failure, the output is a special failure symbol. Sessions have a local state which is erased after the session completes. Besides, each party may have an additional long-term state (composed by long-term signing keys, for example) which is visible to multiple sessions and is not erased after session completion.

The adversarial model is called the "unauthenticated-links model (UM)" and allows the attacker to have full control over the communication channel, thus being able to intercept, delay, inject, drop, or change any message exchanged. In short, it is a fully-capable man-in-the-middle attacker. Besides, the attacker is also allowed to start key exchange sessions and, more importantly, is able to perform all three *session exposure* attacks:

- **Session-state reveal**: targets a still incomplete session. The adversary learns the state of that particular session (not including any long-term secrets accessible to all sessions, such as long-term signing keys).
- **Session-key queries**: targets a complete session and allows the adversary to learn its corresponding session key.
- **Party corruption**: the attacker learns all information possessed by the party (including long-term secrets accessible to all sessions, such as long-term signing keys).

An important concept in this model is *session expiration.* When a session expires, the attacker is not allowed to perform session-state reveal or session-key queries, although is fully able to corrupt a party. A key-exchange protocol which is secure even after a party corruption is said to enjoy *perfect forward secrecy* (PFS). Another relevant concept is the one of matching session.

[1] This security notion was originally introduced in [8]. The main difference between [8] and [9] is that in the former there was an implicit requirement that the identities of the parties must be known to each other beforehand, while the latter attains a more realistic (internet-oriented) scenario where the identities of the parties are not initially known and only becomes known after the protocol run evolves (this model is called the "post-specified peer model" and is the one used in our proposal).

Definition 5. *Let (P, sid) be a complete session with public output (P, sid, Q). The session (Q, sid) is called the matching session of (P, sid) if either:*

1. *(Q, sid) is not completed; or*
2. *(Q, sid) is completed and its public output is (Q, sid, P).*

Finally, the actual concept of SK Secure relies on the attacker's ability of distinguishing a session key from random. This is done through the *test session* game that allows the attacker to choose any session which has not been *exposed* (by any of the session exposure attacks above) nor its matching session, and runs the following game used in the formal SK secure definition.

Game 1 (Test Session) *Let $\mathcal{U}$ be an adversary of the key exchange protocol π. In the test-session game, the key exchange protocol oracle toss a coin $b \leftarrow \{0, 1\}$ and returns a valid session key k if $b = 0$ or returns a sequence of random bits if $b = 1$. The experiment finishes by the adversary $\mathcal{U}$ outputting $b' \in \{0, 1\}$, a guess on the value of b.*

Definition 6 (SK Secure). *A key exchange protocol π is SK Secure in the post-specified peer model with unauthenticated links if the following holds for any adversary:*

1. *π satisfies that both uncorrupted parties output the same session key.*
2. *The probability that $\mathcal{U}$ guesses b correctly in Game 1 is $\frac{1}{2} + \epsilon$, where ϵ is a negligible fraction in the security parameter.*

Having provided this overview on SK Security, we can finally prove the protocol described in Section 4 attains such a security notion.

Theorem 1. *The key exchange protocol described in Section 4 is SK Secure in the post-specified peer model with unauthenticated links assuming that:*

1. *The key exchange protocol described in Section 4 satisfies that both uncorrupted parties output the same session key;*
2. *CAKE scheme is IND-CPA secure;*
3. *SIG and MAC are existentially unforgeable under chosen message attack and that the function f is a secure pseudorandom function.*

Proof. The proof follows Theorem 6.1 of [33]. The first item is about the correctness of the scheme and boils down to ensure that both parties derive the same session key. This is guaranteed by the correctness of the underlying key encapsulation mechanism (CAKE) and the unforgeability of the signature scheme (see third item below) required to ensure that the key corresponds to the decapsulation of the given ciphertext. As in [33], we remark that the security of the MAC and the pseudorandom function are not needed for such a correctness proof. The second item is achieved by Proposition 1 (or Proposition 2 in the QROM), which actually demonstrates that CAKE is IND-CCA secure, a (stronger) security notion that automatically ensures IND-CPA as well. The third item is achieved by construction, i.e. by selecting a MAC and a signature scheme that are EUF-CMA and the function f that is a secure pseudorandom function. □

Remark on Perfect Forward Secrecy. Key exchange protocols based on asymmetric encryption, such as key-transport protocols, are usually not able to achieve PFS. This happens because if a party is compromised, then its long-term encryption keys are also compromised, allowing the adversary to recover past session keys by decrypting previously exchanged ciphertexts. We remark that this is not the case of the protocol described in Section 4 given the fact we use ephemeral asymmetric encryption keys. Hence, since they are part of the session state, they will also be erased in an event of session expiration. Signing keys are the actual long-term keys in our proposal and their leakage does not affect previous sessions. The same argument holds for CAKE as long as the ephemeral encryption keys are guaranteed to be erased after key-exchange completion.

6 Practical Security Assessment

This section discusses the practical security aspects of our proposal.

6.1 Hard Problems and Security Reduction

Let $\mathcal{R}$ be the ring $\mathbb{F}_2[x]/\langle x^r - 1\rangle$. For every $h \in \mathcal{R}$ and any positive integer t, let $\mathcal{E}(h,t)$ denote the uniform distribution over $\{e_0 + e_1 h \mid e_0, e_1 \in \mathcal{R}, \mathsf{wt}(e_0) + \mathsf{wt}(e_1) = t\}$. For any positive integer w, let $\mathcal{K}(w)$ denote the uniform distribution over $\{h_1 h_0^{-1} \mid h_0, h_1 \in \mathcal{R}, \mathsf{wt}(h_0) + \mathsf{wt}(h_1) = w\}$.

The KEM of §3 is secure as long as both distributions $\mathcal{E}(h,t)$ and $\mathcal{K}(w)$ are computationally indistinguishable from the uniform distribution over $\mathcal{R}$. From the practical viewpoint, this means that r, w, t must be chosen such that the following two problems are intractable:

Problem 1. Given $s, h \in \mathcal{R}$, find $e_0, e_1 \in \mathcal{R}$ such that $\mathsf{wt}(e_0) + \mathsf{wt}(e_1) = t$ and $e_0 + e_1 h = s$.

Problem 2. Given $h \in \mathcal{R}$, find $h_0, h_1 \in \mathcal{R}$ such that $\mathsf{wt}(h_0) + \mathsf{wt}(h_1) = w$ and $h_1 + h_0 h = 0$.

Problem 1 and Problem 2 are respectively the problems of decoding t errors and finding a codeword of weight w in an arbitrary quasi-cyclic code of dimension r and length $n = 2r$.

In the current state of the art, the best known techniques for solving those problems are variants of Prange's Information Set Decoding (ISD) [36].

6.2 Information Set Decoding

The best asymptotic variant of ISD is due to May and Ozerov [28], but it has a polynomial overhead which is difficult to estimate precisely. In practice, the BJMM variant [2] is probably the best for relevant cryptographic parameters. The work factor for classical (*i.e.* non quantum) computing of any variant $\mathcal{A}$

of ISD for decoding t errors (or finding a word of weight t) in a binary code of length n and dimension k can be written

$$\mathrm{WF}_{\mathcal{A}}(n,k,t) = 2^{ct(1+o(1))}$$

where c depends on the algorithm, on the code rate $R = k/n$ and on the error rate t/N. It has been proven in [42] that, asymptotically, for sublinear weight $t = o(n)$ (which is the case here as $w \approx t \approx \sqrt{n}$), we have $c = \log_2 \frac{1}{1-R}$ for all variants of ISD.

In practice, when t is small, using 2^{ct} with $c = \log_2 \frac{1}{1-R}$ gives a remarkably good estimate for the complexity. For instance, non asymptotic estimates derived from [18] gives $\mathrm{WF}_{\mathrm{BJMM}}(65542, 32771, 264) = 2^{263.3}$ "column operations" which is rather close to 2^{264}. This closeness is expected asymptotically, but is circumstantial for fixed parameters. It only holds because various factors compensate, but it holds for most MDPC parameters of interest.

Exploiting the Quasi-Cyclic Structure. Both codeword finding and decoding are a bit easier (by a polynomial factor) when the target code is quasi-cyclic. If there is a word of weight w in a QC code then its r quasi-cyclic shifts are in the code. In practice, this gives a factor r speedup compared to a random code. Similarly, using Decoding One Out of Many (DOOM) [39] it is possible to produce r equivalent instances of the decoding problem. Solving those r instances together saves a factor $\sqrt{r}$ in the workload.

Exploiting Quantum Computations. As commented in [4], Grover's algorithm fully applies to Prange algorithm. Effectively, this halves the above asymptotic exponent for Prange algorithm. Later, it was proven in [23] that more involved variants of ISD could achieve a better exponent but also the improvement was disappointingly away from the factor 2 that could be expected. In the sequel, we will estimate the quantum security by dividing the classical exponent by two. This is probably conservative but a more precise evaluation would not be significantly different.

Practical Parameter Selection. We denote $\mathrm{WF}(n,k,t)$ the workfactor of the best ISD variant for decoding t errors in a binary code of length n and dimension k. In the following we will consider only codes of transmission rate 0.5, that is length $n = 2r$ and dimension r. In a classical setting, the best solver for Problem 1 has a cost $\mathrm{WF}(2r,r,t)/\sqrt{r}$ and the best solver for Problem 2 has a cost $\mathrm{WF}(2r,r,w)/r$. As remarked above, with $\mathrm{WF}(2r,r,t) \approx 2^t$ we obtain a crude but surprisingly accurate, parameter selection rule. To reach λ bits of quantum security, we choose w, t and r such that

$$\lambda \approx \frac{t - \frac{1}{2}\log_2 r}{2} \approx \frac{w - \log_2 r}{2}. \quad (1)$$

6.3 Defeating the GJS reaction attack

Both CAKE and the authenticated key exchange protocol described in Section 4 requires ephemeral KEM key pair, i.e. a KEM key generation is performed for each key exchange. As a result, the GJS reaction attack is inherently defeated: a GJS adversary would have (at most) a single opportunity to observe decryption, thus not being able to create statistics about different error patterns. We note that, for efficiency purposes, an initiator may want to precompute KEM key pairs before engaging in key exchange sessions. We remark that policies to securely store the pregenerated KEM key pair must be in place, in order to avoid that an adversary access a KEM key pair to be used in a future communication.

6.4 How to Choose MDPC Parameters

If we denote λ the (quantum) security parameter, then both t and w must be close to 2λ, as in (1). In addition, to ensure decoding, we expect the product tw to grow as $r \log r$. Putting everything together we obtain

$$\begin{cases} t \approx 2\lambda + \log_2(2\lambda) \\ w \approx 2\lambda + 2\log_2(2\lambda) \end{cases}$$

and r will grow as $\lambda^2 / \log \lambda$. The exact value of r needs to be checked, by simulation, and increased to a point where the decoding failure rate is acceptable.

Finally, we choose r such that 2 is primitive modulo r. First, this will force r to be prime, thwarting the so-called squaring attack [27]. Also, it implies that $x^r - 1$ only has two irreducible factors (one of them being $x - 1$). This is an insurance against an adversary trying to exploit the structure of $\mathbb{F}_2[x]/\langle x^r - 1\rangle$ when $x^r - 1$ has small factors, other than $x - 1$.

The parameters suggested in Table 1 consider the security attacks discussed in Section 6. In addition, the block size r is chosen so that state-of-the-art bit flipping decoding has a failure rate not exceeding 10^{-6}. The last column shows the public and private key size which are both n bits long.

λ	n_0	n	r	w	t	key size (bits)
128	2	65,542	32,771	274	264	65,542
96	2	39,706	19,853	206	199	39,706
64	2	20,326	10,163	142	134	20,326

Table 1: QC-MDPC suggested parameters for λ bits of quantum security.

7 Conclusion

This paper introduced CAKE, an IND-CCA secure key encapsulation mechanism (KEM) based on QC-MDPC codes. CAKE uses ephemeral keys and therefore inherently defeats the recent GJS attack [17]. Since key generation is performed for every key exchange, we devised an efficient MDPC key generation, which is much faster than the original method proposed for the MDPC McEliece encryption scheme [32].

Performance-wise, CAKE offers a competitive solution. The public key is n bits long and the cryptogram is $(n + \ell_H)$ bits long, corresponding to the bandwidth of the first and second messages depicted in Figure 1, respectively. The key generation cost is dominated by two sparse-dense modular polynomial multiplications, and does not require any polynomial inversion as usually seen in code-based cryptosystems. The cost of encapsulation is dominated by four hash computations over n, n, ℓ_H and $2n$ bits respectively (except to the third one, all other hash computations receive sparse vectors as inputs, which can be represented in a compact form to reduce the hash input size) and two dense-dense modular polynomial multiplications. The cost of decapsulation is dominated by two sparse-dense polynomial modular multiplications, a decoding attempt, and four hash computations (again over n, n, ℓ_H and $2n$ bits, respectively). In summary, besides MDPC decoding, CAKE relies on modular polynomial multiplications and hash computations, so we can expect efficient implementations on a wide range of platforms. A comprehensive assessment of implementation aspects will be discussed in the full version of this paper.

CAKE compares well with other post-quantum key-exchange schemes. Comparing to Goppa-based McBits [5], the only other currently known code based KEM, we note that CAKE enjoys public keys whose size is orders of magnitude smaller. From a security perspective, we note that Goppa codes may not be optimal, as evidenced by a distinguisher for certain (i.e., high-rate) Goppa codes [13].

Recent works [22, **?**] have shown that isogenies in supersingular elliptic curves can be used to devise efficient key exchange mechanisms. In particular, those constructions have the benefit of achieving small public key sizes. However, this is a much more recent trend and caution should be exercised as they have not gone through nearly as similar scrutiny as code-based cryptosystems, first appeared almost 40 years ago, have endured.

When comparing with lattice-based schemes, e.g., [7], [1] and [6], CAKE and the lattice-based protocols show some similarities. All of them suffer from decoding failures (lattice schemes usually have a lower failure probability though). Also, the use of ephemeral keys for key exchange is not new in the literature; [1] discusses the security loss inherent to key cache ([14] presents a comprehensive analysis on the security impact of key reuse for Ring-LWE). Besides, they offer unbalanced cost between the parties, what may lead to great flexibility (e.g., in a certain application, the role of Initiator/Responder could be predefined depending on the expected computational power of the parties). In terms of total bandwidth cost, CAKE's traffic requires 2/3 of the traffic presented in [6], but is 1.3 and 3 times larger than that of [7] and [1], respectively. While such comparisons are certainly useful, we point out that lattice-based schemes are not the immediate "competitors" of CAKE, because they are based on a different class of hard problems. We note that the transition to post-quantum cryptography is an unprecedented move, thus, relying on a single, silver-bullet class of cryptographic problems (e.g., lattices) is a very risky strategy, whilst considering a set of well-studied constructions seems a considerably safer choice in the long term.

This paper also presents an SK secure authenticated key exchange protocol based on CAKE, which is suitable for the Internet Key Exchange (IKE), similarly to [33]. We prove that CAKE is IND-CCA secure, and that the authenticated protocol is SK secure. Moreover, we demonstrate that our proposal achieves perfect forward secrecy, despite the fact it is based on asymmetric encryption (key transport schemes with static encryption keys do not attain PFS, for example).

Taking all these considerations into account, we believe that CAKE is a promising candidate for post-quantum key exchange standardization.

References

1. Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. Post-quantum key exchange - a new hope. Cryptology ePrint Archive, Report 2015/1092, 2015. `http://eprint.iacr.org/2015/1092`.
2. Anja Becker, Antoine Joux, Alexander May, and Alexander Meurer. Decoding random binary linear codes in $2^{n/20}$: How 1+1=0 improves information set decoding. In D. Pointcheval and T. Johansson, editors, *Advances in Cryptology - EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 520–536. Springer, 2012.
3. E. Berlekamp, R. McEliece, and H. van Tilborg. On the inherent intractability of certain coding problems (corresp.). *Information Theory, IEEE Transactions on*, 24(3):384 – 386, may 1978.
4. Daniel J. Bernstein. *Grover vs. McEliece*, pages 73–80. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.
5. Daniel J. Bernstein, Tung Chou, and Peter Schwabe. Mcbits: Fast constant-time code-based cryptography. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 8086 LNCS of *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, pages 250–272, 12 2013.
6. Joppe Bos, Craig Costello, Léo Ducas, Ilya Mironov, Michael Naehrig, Valeria Nikolaenko, Ananth Raghunathan, and Douglas Stebila. Frodo: Take off the ring! practical, quantum-secure key exchange from LWE. Cryptology ePrint Archive, Report 2016/659, 2016. `http://eprint.iacr.org/2016/659`.
7. Joppe W Bos, Craig Costello, Michael Naehrig, and Douglas Stebila. Post-quantum key exchange for the tls protocol from the ring learning with errors problem. In *Security and Privacy (SP), 2015 IEEE Symposium on*, pages 553–570. IEEE, 2015.

8. Ran Canetti and Hugo Krawczyk. Analysis of key-exchange protocols and their use for building secure channels. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 453–474. Springer, 2001.
9. Ran Canetti and Hugo Krawczyk. Security analysis of ike's signature-based key-exchange protocol. In *Annual International Cryptology Conference*, pages 143–161. Springer, 2002.
10. Pierre-Louis Cayrel, Gerhard Hoffmann, and Edoardo Persichetti. Efficient implementation of a cca2-secure variant of McEliece using generalized Srivastava codes. In *Proceedings of PKC 2012, LNCS 7293, Springer-Verlag*, pages 138–155, 2012.
11. Julia Chaulet and Nicolas Sendrier. Worst case QC-MDPC decoder for McEliece cryptosystem. In *Information Theory (ISIT), 2016 IEEE International Symposium on*, pages 1366–1370. IEEE, 2016.
12. R. Cramer and V. Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM J. Comput.*, 33(1):167–226, January 2004.
13. Jean-Charles Faugere, Valérie Gauthier-Umana, Ayoub Otmani, Ludovic Perret, and Jean-Pierre Tillich. A distinguisher for high-rate McEliece cryptosystems. *IEEE Transactions on Information Theory*, 59(10):6830–6844, 2013.
14. Scott Fluhrer. Cryptanalysis of ring-lwe based key exchange with key share reuse. Cryptology ePrint Archive, Report 2016/085, 2016. `http://eprint.iacr.org/2016/085`.
15. E. Fujisaki and T. Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In Michael J. Wiener, editor, *Advances in Cryptology - CRYPTO '99*, volume 1666 of *LNCS*, pages 537–554. Springer, 1999.
16. R. G. Gallager. *Low-Density Parity-Check Codes.* PhD thesis, M.I.T., 1963.
17. Qian Guo, Thomas Johansson, and Paul Stankovski. *A Key Recovery Attack on MDPC with CCA Security Using Decoding Errors*, pages 789–815. Springer Berlin Heidelberg, Berlin, Heidelberg, 2016.
18. Yann Hamdaoui and Nicolas Sendrier. A non asymptotic analysis of information set decoding. Cryptology ePrint Archive, Report 2013/162, 2013. `http://eprint.iacr.org/2013/162`.
19. Dan Harkins and Dave Carrel. Rfc 2409: The internet key exchange (ike). *Status: Proposed Standard*, 1998.
20. Stefan Heyse, Ingo Von Maurich, and Tim Güneysu. Smaller keys for code-based cryptography: QC-MDPC McEliece implementations on embedded devices. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 273–292. Springer, 2013.
21. Dennis Hofheinz, Kathrin Hvelmanns, and Eike Kiltz. A modular analysis of the Fujisaki-Okamoto transformation. Cryptology ePrint Archive, Report 2017/604, 2017. `http://eprint.iacr.org/2017/604`.
22. David Jao and Luca De Feo. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In *International Workshop on Post-Quantum Cryptography*, pages 19–34. Springer, 2011.
23. Ghazal Kachigar and Jean-Pierre Tillich. Quantum information set decoding algorithms. In Tanja Lange and Tsuyoshi Takagi, editors, *PQCrypto 2017*, volume 10346 of *LNCS*, pages 69–89. Springer, 2017.
24. Hugo Krawczyk. Sigma: The 'sign-and-mac'approach to authenticated diffie-hellman and its use in the ike protocols. In *Annual International Cryptology Conference*, pages 400–425. Springer, 2003.
25. Tanja Lange. Initial recommendations of long-term secure post-quantum systems. *PQCRYPTO. EU. Horizon*, 2020, 2015.

26. Vadim Lyubashevsky, Chris Peikert, and Oded Regev. *On Ideal Lattices and Learning with Errors over Rings*, pages 1–23. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.
27. Carl Lndahl, Thomas Johansson, Masoumeh Koochak Shooshtari, Mahmoud Ahmadian-Attari, and Mohammad Reza Aref. Squaring attacks on McEliece public-key cryptosystems using quasi-cyclic codes of even dimension. *Designs, Codes and Cryptography*, 80(2):359–377, 2016.
28. Alexander May and Ilya Ozerov. On computing nearest neighbors with applications to decoding of binary linear codes. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology – EUROCRYPT 2015, Part I*, volume 9056 of *LNCS*, pages 203–228. Springer, 2015.
29. R. J. McEliece. A public-key cryptosystem based on algebraic coding theory. *Deep Space Network Progress Report*, 44:114–116, January 1978.
30. Daniele Micciancio. Improving lattice based cryptosystems using the hermite normal form. *Cryptography and lattices*, pages 126–145, 2001.
31. V. S. Miller. Use of elliptic curves in cryptography. In *Advances in cryptology (CRYPTO 85)*, pages 417–426, New York, USA, 1986. Springer-Verlag.
32. R. Misoczki, J.-P. Tillich, N. Sendrier, and P. L.S.M. Barreto. MDPC-McEliece: New McEliece variants from moderate density parity-check codes. In *IEEE International Symposium on Information Theory – ISIT'2013*, pages 2069–2073, Istambul, Turkey, 2013. IEEE.
33. Chris Peikert. Lattice cryptography for the internet. In *International Workshop on Post-Quantum Cryptography*, pages 197–219. Springer, 2014.
34. Chris Peikert. A decade of lattice cryptography. *Foundations and Trends® in Theoretical Computer Science*, 10(4):283–424, 2016.
35. Edoardo Persichetti. Secure and anonymous hybrid encryption from coding theory. In Philippe Gaborit, editor, *Post-Quantum Cryptography: 5th International Workshop, PQCrypto 2013, Limoges, France, June 4-7, 2013. Proceedings*, pages 174–187, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.
36. E. Prange. The use of information sets in decoding cyclic codes. *IRE Transactions*, IT-8:S5–S9, 1962.
37. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM (JACM)*, 56(6):34, 2009.
38. R. L. Rivest, A. Shamir, and L. M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.
39. N. Sendrier. Decoding one out of many. In B.-Y. Yang, editor, *PQCrypto 2011*, volume 7071 of *LNCS*, pages 51–67. Springer, 2011.
40. Karen Seo and Stephen Kent. Security architecture for the internet protocol. *Status: Proposed Standard*, 2005.
41. 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.
42. Rodolfo Canto Torres and Nicolas Sendrier. Analysis of information set decoding for a sub-linear error weight. In Tsuyoshi Takagi, editor, *PQCrypto 2016*, volume 9606 of *LNCS*, pages 144–161. Springer, 2016.
43. Ingo Von Maurich and Tim Güneysu. Lightweight code-based cryptography: QC-MDPC McEliece encryption on reconfigurable devices. In *Proceedings of the conference on Design, Automation & Test in Europe*, page 38. European Design and Automation Association, 2014.

44. Ingo Von Maurich and Tim Güneysu. Towards side-channel resistant implementations of QC-MDPC McEliece encryption on constrained devices. *PQCrypto*, 2014:266–282, 2014.

A Efficiently Sampling Invertible Elements from $\mathbb{F}_2[x]/\langle x^r - 1\rangle$

In this section, we prove that one can efficiently sample an invertible element from $\mathbb{F}_2[x]/\langle x^r - 1\rangle$ by taking any polynomial $h \xleftarrow{\$} \mathbb{F}_2[x]/\langle x^r - 1\rangle$ such that $\mathsf{wt}(h)$ is odd.

Lemma 1. *Let $h \in \mathbb{F}_2[x]$ have even weight. Then h is not invertible modulo $x^r - 1$.*

Proof. We show that $(x-1) \mid h$ by induction on $\mathsf{wt}(h)$. For $\mathsf{wt}(h) = 0$ trivially $(x-1) \mid h$. Assume that $(x-1) \mid h$ whenever $\mathsf{wt}(h) = 2k$ for some $k \geqslant 0$. Now consider any $h \in \mathbb{F}_2[x]$ with weight $\mathsf{wt}(h) = 2(k+1)$, and take two distinct terms x^i, x^j of h such that $i < j$. Define $h' = h - x^i - x^j$, so that $\mathsf{wt}(h') = 2k$. Then $(x-1) \mid h'$ by induction, i.e. $h' = (x-1)h''$ for some $h'' \in \mathbb{F}_2[x]$. Hence $h = h' + x^i + x^j = (x-1)h'' + x^i(x^{j-i}+1) = (x-1)h'' + x^i(x-1)(x^{j-i-1} + \cdots + 1) = (x-1)(h'' + x^i(x^{j-i-1} + \cdots + 1))$, and therefore $(x-1) \mid h$. □

Theorem 2. *Let r a prime such that $(x^r - 1)/(x-1) \in \mathbb{F}_2[x]$ is irreducible. Then any $h \in \mathbb{F}_2[x]$ with $\deg(h) < r$ is invertible modulo $x^r - 1$ iff $h \neq x^{r-1} + \cdots + 1$ and $\mathsf{wt}(h)$ is odd.*

Proof. Take a term x^i of h. Then $\mathsf{wt}(h + x^i) = \mathsf{wt}(h) - 1$ is even, and by Lemma 1 $(x-1) \mid (h + x^i)$. Hence $h \bmod (x-1) = x^i \bmod (x-1) = 1$, meaning that h is invertible modulo $x - 1$.

Now, because $(x^r - 1)/(x-1) = x^{r-1} + \cdots + 1$ is irreducible, if $\deg(h) < r-1$ then $\gcd(h, x^{r-1} + \cdots + 1) = 1$, and if $\deg(h) = r-1$, then $\gcd(h, x^{r-1} + \cdots + 1) = \gcd(h + x^{r-1} + \cdots + 1, x^{r-1} + \cdots + 1) = 1$, since $\deg(h + x^{r-1} + \cdots + 1) < r - 1$. Hence h is invertible modulo $x^{r-1} + \cdots + 1$.

Therefore, the combination of the inverses of h modulo $x - 1$ and modulo $x^{r-1} + \cdots + 1$ via the Chinese remainder theorem is well defined, and by construction it is the inverse of h modulo $(x-1)(x^{r-1} + \cdots + 1) = x^r - 1$. □

Corollary 1. *One can efficiently sample an invertible element from $\mathbb{F}_2[x]/\langle x^r - 1\rangle$ by taking any polynomial $h \xleftarrow{\$} \mathbb{F}_2[x]/\langle x^r - 1\rangle$ such that $\mathsf{wt}(h)$ is odd.* □

B Underlying Cryptosystem and Properties

In this section, we briefly discuss the cryptosystem underlying CAKE, and its properties.

The scheme follows almost completely the "classical" McEliece framework, but with a twist. In fact, as mentioned in Section 3, we interpret McEliece encryption as having the message conveyed in the error vector, rather than the codeword; that is, if m is the plaintext and u the randomness we have $\textsc{Enc}(m,u) = uG + m$ rather than $mG + u$. This technique is not new, following the lines of Micciancio's work in [30] and having already been used in a code-based scheme by Cayrel et al. in [10]. Furthermore, we are able to guarantee the γ-spread property for it. Recall the definition of γ-spread PKE as in Definition 4. It is easy to show that this variant of McEliece satisfies this definition, since it was proved in [10] that the scheme is γ-uniform for $\gamma = 2^{-r}$, where r is the code dimension as per our notation (more in general, $\gamma = q^{-r}$ for a cryptosystem defined over $\mathbb{F}_q$).

C Standard KEM Security Definitions

Below we present the IND-CCA security definition for a KEM.

Definition 7. *The adaptive chosen-ciphertext attack game for a KEM proceeds as follows:*

1. *Query a key generation oracle to obtain a public key pk.*
2. *Make a sequence of calls to a decryption oracle, submitting any string C of the proper length. The oracle will respond with $\textsc{Decaps}(sk, C)$.*
3. *Query an encryption oracle. The oracle runs $\textsc{Encaps}(pk)$ to generate a pair $(\tilde{K}, \tilde{C})$, then chooses a random $b \in \{0,1\}$ and replies with the "challenge" ciphertext $(K^*, \tilde{C})$ where $K^* = \tilde{K}$ if $b = 1$ or K^* is a random string of length ℓ otherwise.*
4. *Keep performing decryption queries. If the submitted ciphertext is C^*, the oracle will return $\perp$.*
5. *Output $b^* \in \{0,1\}$.*

The adversary succeeds if $b^* = b$. More precisely, we define the advantage *of $\mathcal{A}$ against KEM as*

$$\mathsf{Adv}_{KEM}^{IND-CCA}(\mathcal{A}, \lambda) = \left| \mathsf{Pr}[b^* = b] - \frac{1}{2} \right|. \tag{2}$$

We say that a KEM is secure if the advantage $\mathsf{Adv}_{KEM}^{IND-CCA}$ of any polynomial-time adversary $\mathcal{A}$ in the above CCA attack model is negligible.

The IND-CPA security notion is defined exactly as above, except that the adversary is not allowed to perform any decryption queries.