Mathematics

https://www.aimspress.com/journal/Math

AIMS Mathematics, 10(8): 17423–17458.
DOI: 10.3934/math.2025779
Received: 27 March 2025
Revised: 09 July 2025
Accepted: 22 July 2025
Published: 01 August 2025

Survey

Quantum-resistant authentication: Securing identity and data against quantum threats

Ana I. González-Tablas[1] **and María Isabel González Vasco**[2,*]

[1] Universidad Carlos III de Madrid, Dpto. de Informática, COSEC Lab. Avda. de la Universidad, 30. 28911 Leganés, Madrid, España

[2] Universidad Carlos III de Madrid, Dpto. de Matemáticas, Avda. de la Universidad, 30. 28911 Leganés, Madrid, España

* **Correspondence:** Email: mariaisabel.gonzalez@uc3m.es.

Abstract: Quantum computing presents a significant threat to traditional cryptographic protocols, particularly those relying on the hardness of factoring large integers and computing discrete logarithms, as quantum algorithms can solve these problems in polynomial time. With the advent of quantum computers capable of breaking current cryptographic schemes, the need for quantum-resistant cryptographic mechanisms has become increasingly urgent. In this paper, we focused on authentication, a critical security property, and reviewed various classical cryptographic techniques to this aim and their potential to resist quantum attacks. We present a (non-exhaustive) overview of the state of the art in quantum-resistant authentication, particularly through symmetric cryptography, digital signatures, and hardware-based approaches such as physical unclonable functions (PUFs). By offering a clear and focused analysis, we used this survey to bridge the gap between diverse academic communities and provide a foundation for future research in post-quantum authentication.

Keywords: authentication; quantum key distribution; post-quantum cryptography; signature schemes; symmetric cryptography; physical unclonable functions

1. Introduction

With the upcoming development and deployment of quantum computers of sufficient capacity, many systems will need to upgrade the procedures they use to provide authentication services to quantum-resistant mechanisms. To ensure a successful transition to quantum-resistant (often called post-quantum) tools, it is essential to adopt a multidisciplinary approach by involving cryptography and security experts from across the academic community, including theoretical computer scientists, mathematicians, physicists, and system engineers. However, various academic communities often employ distinct terminologies, making it challenging to facilitate effective interaction.

Quantum computing poses a significant threat to numerous cryptographic implementations. This emerging paradigm challenges the conventional boundaries of complexity classes, as the development of novel quantum algorithms enables the resolution of certain problems in quantum polynomial time, whereas their classical* counterparts require exponential time. On top of the famous algorithms for period finding and search, due to Peter Shor [1] and Lov Grover [2], a number of novel algorithmic techniques and developments help us grasp which computational tasks are effectively simplified throught quantum technologies and how fast can they be performed in practice [3]. For the cryptographic community, the most tangible impact of quantum computing is the loss of security for cryptographic schemes based on the hardness of factoring large integers or computing discrete logarithms in cyclic groups, as these problems become efficiently solvable by quantum adversaries. Consequently, new cryptographic designs are being proposed and analyzed, drawing on problems from both well-established and previously unexplored areas of mathematics. This rapidly evolving research field, known as post-quantum cryptography, aims to develop cryptographic systems resilient to quantum attacks. On the other hand, new adversarial models must be developed to accurately capture the capabilities of adversaries equipped with quantum technologies. Constructing such models is considerably more complex than merely considering adversaries that can efficiently solve discrete logarithm or factorization problems. It is essential to account for the fundamentally different properties of quantum systems, such as those implied by superposition, the no-cloning theorem, and the impossibility of rewinding. These quantum-specific characteristics significantly influence the interactions among the various entities in a cryptographic protocol and must be rigorously incorporated into security analyses (see, for instance, [4]).

Authentication is a fundamental security property that guarantees the legitimacy of an entity, whether a user, device, or system, as well as the integrity and origin of transmitted data. Various protocols and techniques, both interactive and non-interactive, are routinely deployed to facilitate authentication across a wide range of practical applications, including many of critical importance. A notable example is the authentication of software updates and firmware, which represents a common target for adversarial attacks, particularly when systems require upgrades to enhance security.

In this survey, we seek to bridge the relevant communities by offering a comprehensive overview of the current state of the art in quantum-resistant authentication. We focus on three major categories of tools: Those based on symmetric cryptography, those utilizing asymmetric or public-key cryptography, and those built on so-called physical unclonable functions, a notable example of hardware-based authentication as a versatile resource. Although we present these areas separately, we emphasize that these types of tools are often combined in real-world applications.

The scope of this survey is confined to constructions based on classical (i.e., non requiring quantum resources) technologies, with a particular emphasis on the mathematical tools and arguments that underpin them. While we do not intend to provide an exhaustive analysis, our objective is to present an overview of the most promising techniques and methodologies in this field.

Related work. Another recent survey on quantum-resistant authentication can be found in [5]. While that work provides an exhaustive list of various quantum-resistant solutions, our approach is more limited and in a way, selective, as we focus on conveying the underlying mathematical ideas behind

*Throughout this text, the term "classical" will refer to non-quantum technologies.

some of the most relevant constructions in the field. Furthermore, we have considered authentication in a broad sense, whereas other researchers typically focus on quantum-resistant adaptations of specific protocols used for secure communication, such as TLS (see [6]). In addition, in this paper, we have limited our focus to authentication techniques that can be implemented using classical technologies, but are potentially resistant to attacks that exploit quantum computing resources. However, using quantum techniques for authentication (most often, of entities, i.e., identification) has been extensively researched ([7–11]). For a survey on the topic, we refer the interested reader to [12].

Paper roadmap. We begin by reviewing some notions and ideas needed to make this paper self-contained in Section 2. To this aim, we first revisit some mathematical notions involved in post-quantum constructions, and then focus on basic concepts related to authentication. Further, we devote Section 3 to explain the major tools from symmetric cryptography typically used for post-quantum authentication. Then, we move on to digital signature schemes in Section 4, where the rationale of several post-quantum public key signature scheme proposals is briefly explained and some recent relevant examples depicted. Finally, Section 5 is devoted to hardware based authentication, particularly focusing on the use of physical unclonable functions. We conclude our overview with some final remarks in Section 6.

2. Background

This section includes some notions and examples that will help the reader gain understanding on the tools available and challenges to tackle when trying to design quantum-resistant authentication methods.

2.1. The mathematics of post-quantum cryptography

The quest for mathematical problems whose computational difficulty remains unaffected by advancements in quantum technologies presents significant and highly complex research challenges. The area has attracted a lot of interest in the last decade, not only due to the competition launched in 2016 by the US National Institute for Standars and Technology (NIST) with the aim of standarizing post-quantum cryptographic constructions. This process was structured through a first call for proposals for key encapsulation mechanisms (KEMs) and digital dignature schemes (DSSs). As both public key encryption schemes and key establishment protocols can be derived from KEMs, the NIST call ambitioned to have a full suite of post-quantum cryptographic primitives standarized by the end of the process. The competition has gone through multiple evaluation rounds to assess security, efficiency, and practicality of the submited proposals. In 2022, NIST selected four primary constructions (CRYSTALS-Kyber [13] for key encapsulation, and CRYSTALS-Dilithium [14], FALCON [15], and SPHINCS$^+$ [16] for digital signatures). All of these are based on lattice problems, except from SPHINCS$^+$, which is a hash-based signature. The process concluded in March 2025 with the selection of a code-based construction, HQC, as the fifth algorithm to be standarized (see the final NIST report [17]). In addition, a subsequent process was opened in 2022 and is running in order to standarize further signature schemes.

While indeed lattice based cryptography is one of the major actors in this field, the mathematics of post-quantum cryptography is diverse, encompassing several problem domains. Let us provide a brief overview of the key research avenues in this pursuit.

Lattice-based cryptography. The hardness of solving different problems in discrete lattices has since the early days of public key cryptography been exploited for cryptographic constructions. Given a basis for a certain discrete lattice, classical problems such as the *Short Integer Solution Problem* (SIS), *Shortest Vector Problem* (SVP), or *Closest Vector Problem* (CVP) have been used for many cryptographic constructions.

Lattice-based Problems

Short Integer Solution Problem (SIS) Given two integers n, m, a prime number q and a matrix A defined by m (column) vectors selected uniformly at random from $\mathbb{Z}_q^n$, find a *short* nonzero vector $v \in \mathbb{Z}^m$ such that $Av = 0 \mod q$.
Shortest Vector Problem (SVP): Given a field $\mathbb{K}$, a lattice Λ of dimension $n \in \mathbb{N}$ over $\mathbb{K}$ is the set of all integer linear combinations of a given set of linearly independent vectors $\{b_1, \ldots, b_n\}$. Find the nonzero vector $\lambda \in \Lambda$ with minimal norm.

Closest Vector Problem (CVP): Given a field $\mathbb{K}$, a lattice Λ of dimension $n \in \mathbb{N}$ over $\mathbb{K}$ is the set of all integer linear combinations of a given set of linearly independent vectors $\{b_1, \ldots, b_n\}$. Given a target vector $v \notin \Lambda$, find the lattice vector $\lambda \in \Lambda$ closest to v.

Learning with errors (LWE) – search : Given a matrix $A \in \mathbb{Z}_q^{m\times n}$ and a noisy vector $b \in \mathbb{Z}_q^m$, find $s \in \mathbb{Z}_q^n$ such that $As + e = b$ (for some bounded noise vector $e \in \mathbb{Z}_q^m$).

In a nutshell, given a field $\mathbb{K}$ a lattice Λ of dimension $n \in \mathbb{N}$ over $\mathbb{K}$ can be defined as the subset of all integer linear combinations of a given set of linearly independent vectors $\{b_1, \ldots, b_n\}$ (e.g., a basis for Λ). Once a norm is defined, a natural question is to ask for the shortest vector $\lambda \in \Lambda$ (i.e., the vector of minimal norm) and also to try to compute the closest vector to a given $v \notin \Lambda$, which is how, respectively, the SVP and the CVP are defined. In turn, SIS is defined as the problem of finding a short non-zero vector given a random matrix; in a seminal work, Ajtai [18] presented a family of one-way functions based on this problem and showed the relation to SIS to SVP (SIS is secure on average if a certain SVP is hard in a worst-case sense).

More recently, the so-called *Learning With Errors* (LWE) problem, has taken a starring role in this field (see the seminal paper by Regev, [19]). The idea behind it is very intuitive; fixing two natural numbers n, m and given $\mathbb{Z}_q$, the ring of integers modulo $q > 2$, a matrix $A \in \mathbb{Z}_q^{m\times n}$ and a "noisy" vector $b \in \mathbb{Z}_q^m$, the search version of the problem is the problem of finding a secret vector $s \in \mathbb{Z}_q^n$ such that

$$As + e = b,$$

for some bounded noise vector $e \in \mathbb{Z}_q^m$. Note that in this case, the considered lattice is defined, taken the rows of A as a basis. In turn, a decisional version of the problem can be defined as the problem of telling apart random samples from samples generated as $As + e$ — for fixed s and random (A, e). It was soon realized that more efficient cryptographic constructions could be derived choosing a different algebraic structure instead of $\mathbb{Z}_q$, namely, for some suitable rings, single equations (instead of systems) could yield more samples; the corresponding problem is called *Ring Learning With Errors* (RLWE) and hardness results on these where established by Lyubashevsky et al [20]. A nice survey on these problems can be found in [21].

One of the oldest cryptographic constructions based on lattices is the NTRU encryption scheme [22], which has been intensely revisited due to the NIST competition. While NTRUE did not make it to the end of the competition, three of the four constructions selected in 2022 by NIST are also lattice-based (namely, Kyber, Dilithium, and FALCON). This has propelled the search for both classical and quantum algorithms for solving the related problems (see, for instance [23]). Note that both Dilithium and Kyber have been published as FIPS 203 and FIPS 204 standards (see ML-KEM [24] and ML-DSA [25]), while FALCON is expected to be published as FN-DSA (future FIPS 206).

Code-based cryptography. Code-based constructions are also very relevant in the post-quantum scenario. Broadly speaking, most are built using the hardness of decoding random linear error-correcting codes, typically formalized as a *General Decoding Problem* (GDP) or as a *Syndrome Decoding Problem* (SDP). A related problem is that of *Code Equivalence* (CEP), the question of whether two given linear or nonlinear codes are equivalent under a specified group of transformations.

Decoding Problems

General Decoding Problem (GDP): Given a generator matrix G for an $[n, k]$ code C over $\mathbb{F}_q$, a non negative integer t_0 and a vector $w \in \mathbb{F}_q^n$, find a codeword $\hat{w} \in C$ with $\delta(w, \hat{w}) \leq t_0$.

Syndrome Decoding (SDP): Given an $(n-k) \times n$ matrix H over $\mathbb{F}_q$, a vector $s \in \mathbb{F}_q^{n-k}$, and a nonnegative integer t_0; find a vector $w \in \mathbb{F}_q^n$, with t_0 nonzero entries and such that $Hw^T = s^T$.

Code Equivalence Problem (CEP): Let $G_1, G_2 \in \mathbb{F}_q^{k\times n}$ be two generator matrices for two linearly equivalent codes C_1 and C_2. Find two matrices over $\mathbb{F}_q$ making this linear equivalence explicit, i.e., S, an $k \times k$ invertible matrix and Q an $n \times n$ matrix, such that

$$G_2 = SG_1Q.$$

In a nutshell, given $n, q \in \mathbb{N}$, a q-ary linear code C is nothing but a subspace of $\mathbb{F}_q^n$, if it has dimension $k \in \mathbb{N}$, we say that C is a q-ary $[n, k]$ code. If G is an $n \times k$ matrix whose rows constitute a basis for C, we say that G is a generator matrix. Dually, if H is an $(n-k) \times n$ matrix such that $G \cdot H^T = 0$, we say that H is a parity check matrix for the code C. The error correction capability t of a code is the maximum number of transmission mistakes that can be tolerated without interfering with the decoding process. Typically, there is a notion of "distance" (δ) defined in order to make this decoding capacity explicit; if the received word w is close enough to the actually transmitted word $\hat{w}$, i.e., $\delta(w, \hat{w}) \leq t$, then it is posible to decode it correctly. However, decoding algorithms typically need to exploit certain concrete characteristics of the concrete code in use; decoding with respect to a random linear code is computationally very hard. To embed a trapdoor in coding-based cryptography, one typically begins with a well-structured secret code C and applies a linear transformation to obtain a code C' that should be indistinguishable from a random code.

The first well-known construction in this scenario is the so-called McEliece cryptosystem, first proposed in 1978 [26]. It remains essentially unbroken despite decades of research. McEliece's original idea was to encrypt messages as codewords of a certain linear code with error-correcting capability, to which a randomly chosen error was added, following a design similar to lattice-based constructions. The public key consists of a somewhat masked generator matrix of the code, while the secret key is the set of information necessary to execute an efficient decoding algorithm capable of removing the introduced errors. Assuming that the generator matrix of the code (which consitutes the public key) is indistinguishable from a random matrix, the adversary is faced with a generic decoding problem, for which no efficient algorithms are known. BIKE [27], Classic McEliece [28], and HQC [29] are three code-based constructions that reached Round 4 of the aforementioned NIST competition. As previously mentioned, HQC emerged as the winner by crossing the finish line.

Multivariate cryptography. Multivariate cryptography relies on the difficulty of solving systems of nonlinear (typically, quadratic) polynomial equations over finite fields. This class of cryptosystems offers fast encryption and decryption but has historically been challenging to design securely. This area is believed to have potential for post-quantum cryptography.

Multivariate Quadratic Problems

Multivariate Quadratic (MQ): Let $m, n, q \in \mathbb{N}$. Consider m quadratic polynomials in n variables over a finite field $\mathbb{F} = \mathbb{F}_q$, given by

$$\forall k = 1, \ldots, m;\ p^k(x_1, \ldots, x_n) = \sum_{i=1}^{n} \sum_{j=1}^{n} p_{ij}^k x_i x_j + \sum_{i=1}^{n} p_i^k x_i + p_0^k$$

with

$$p_{ij}^k, p_i^k, p_0^k \in \mathbb{F}, \forall i, j \in \{1, ..., n\}.$$

Then, the MQ problem associated with the above collection is to find a common root for the m polynomials, i.e., a vector $\vec{x} \in \mathbb{F}^n$ such that

$$p^1(\vec{x}) = \cdots = p^m(\vec{x}) = 0.$$

Informally, a *Multivariate Quadratic* (MQ) problem is the problem of finding a common root for several multivariate quadratic polynomials. It is well known that this problem belongs to the complexity class NP-hard, meaning that there is evidence suggesting that some instances of this problem will be computationally infeasible to solve within a reasonable time, even in its simplest form over a two-element field. However, in order to securely generate a large number of cryptographic keys, it is necessary to find a large number of problem instances that are always hard to solve. Thus, worst-case complexity serves only as a relative indicator. It is conjectured that, when the parameters of an MQ problem are chosen at random (in some sense), its difficulty will be high, supported by results suggesting that the average-case difficulty of this problem is also very high. However, there are no clear guidelines for selecting the involved polynomials with sufficient security guarantees, and less so if one needs to select them so that there is a close relation with an easy system of equations that can be exploited as a trapdoor for enabling decryption or verification of signatures. This is the main challenge in designing cryptographic proposals based on this paradigm, sadly, many promising constructions have through the years been proven insecure.

Examples of celebrated constructions in this area are the seminal proposals by Patarin et al. [30], and, more recently, the Rainbow signature scheme [31] or the new signature schemes MAYO, QR-UOV, SNOVA, and UOV [32], all advancing to the second round of a new NIST competition to standarize post-quantum signatures.

Isogeny-based cryptography. Informally, an isogeny is a map connecting two algebraic curves which is surjective and preserves the underlying group structure (see [33] for formal definitions). In a nutschell, the idea behind isogeny based cryptography is to mimic so-called Diffie-Hellman like constructions using isogenies instead of exponentiation. Two curves connected through an isogeny are said to be isogenous. Given two isogenous elliptic curves over a finite field, computing an isogeny between them, the so-called *General Isogeny Problem* (GIP), is assumed to be a very hard computational problem (even if quantum resources are at hand). However, proving that two such curves are isogenous is possible in (classical) polynomial time. A very interesting problem motivated by practical applications is finding ways to prove knowledge of an isogeny between two curves without actually revealing it [34]. Related problems are that of finding a chain of isogenies connecting two curves (*Isogeny Path Finding Problem* (IPF) or the problem of fully describing the set of structure-preserving transformations of a given curve (known as the *Endomorphism Ring Computation Problem* (ERCP)).

Isogeny-based Problems

General Isogeny Problem (GIP): Given two elliptic curves E and E' (not necessarily supersingular), find an explicit isogeny between them.

Isogeny Path Finding Problem (IPFP): Given two elliptic curves E and E' over a finite field $\mathbb{F}_q$, determine a sequence of isogenies (an isogeny path) that connects E to E'.

Endomorphism Ring Computation Problem (ERCP): Given a supersingular elliptic curve E, compute its endomorphism ring $\mathrm{End}(E)$.

The best known proposals along this line are SIDH (for key exchange) and SIKE (for key encapsulation) [35], both rendered insecure by the attack from [36]. Still, other proposals, such as CSIDH [37] remain unbroken, and are attracting a lot of interest. The fact that isogeny-based problems enable us to somehow mimic non-interactive key exchange à la Diffie-Hellman (thus the sufix "DH" in the acronyms SIDH and CSIDH) makes this research avenue very interesting. Moreover, in the NIST signature contest, SQISign [38], a signature scheme where security is based on the knowledge of a secret elliptic curve isogeny, has been selected to move forward to the second evaluation round.

In search for hard problems. There are other areas of mathematics where different problems are explored to build quantum-resistant constructions upon (see, for instance, recent constructions using group actions [39], or the so-called *Semi Direct Discrete Logarithm Problem*, see the proposal from [40] and the cryptanalysis from [41]). However, it is worth noting that identifying and understanding which mathematical problems are hard enough in a post-quantum sense is not an easy task. In an attempt to identify promising research avenues, the *Hidden Subgroup Problem* (HSP) is understood as a generic formulation englobing many such potentially hard problems. HSP can be seen as a way to understand the power of quantum algorithms and the limits of Shor's algorithm in group theoretical language.

Hidden Subgroup Problem

Hidden Subgroup Problem (HSP): Let G be a known group, $H \leq G$ and let $f : G \to S$ be a function from G to a known set S such that for all $g_1, g_2 \in G$,

$$f(g_1) = f(g_2) \iff g_1 H = g_2 H.$$

Describe H.

An informal formulation of the HSP may be given as follows: let G be a finitely generated group, S a finite set, and consider an efficiently computable function $f : G \to S$ such that f is constant and distinct on left cosets of a subgroup $H \leq G$ of finite index, identifying a generating set for H. Many problems used in cryptography may be reduced to HSP; actually, Shor's [1] polynomial-time quantum algorithms for the *Integer Factorization Problem* and *Discrete Logarithm Problem* rely on a polynomial-time quantum algorithm for HSP in finite cyclic groups and groups of the form $\mathbb{Z}_p \times \mathbb{Z}_p$ for prime p. There are efficient quantum algorithms for HSP for all finite abelian groups and for a few classes of finite non-abelian groups. See [42] for a full survey.

2.2. *Basics on Authentication*

In this section, we present the basic concepts necessary for understanding authentication and briefly explore some mechanisms by which it can be implemented using cryptographic techniques.

Authentication is defined as "a process that provides assurance of the source and integrity of information in communications sessions, messages, documents or stored data or that provides assurance of the identity of an entity interacting with a system" [43]. Authentication serves as a critical barrier against unauthorized access, protecting not only personal data but also the integrity of entire systems. It usually forms the first line of defense against a multitude of cyber threats. The above definition of authentication, in line with common understanding, distinguishes two main authentication services: entity authentication and data authentication. Next, a high-level definition and basic working details of these services are presented to help understand the context in which the previously introduced mathematical problems need to be applied.

Entity authentication. It is the process by which an entity, called the Relaying Party *RP*, verifies the identity of a user or system, called the Prover *Pr*, usually as a requirement before granting the Prover the right to access certain resources. This typically involves the use of a credential *Cred* such as a password, a shared secret or cryptographic key, biometrics (in the case of human entities), or a secure hardware-based token. In general, all entity authentication methods currently used in digital systems involve the use of cryptography, either to securely create or transmit credentials or to avoid the predictability of one-time secrets, among other purposes.

Entity authentication typically involves two phases, corresponding to two separate processes: Enrollment and authentication (see Figure 1). During the enrollment phase, the Relying Party registers ID_{Pr}, the identity of the Prover, and both convey a credential *Cred* for the Prover. The

Relying Party registers the information needed to verify the credential along with the identity of the Prover. The Prover is required to keep the credential, which usually contains some secret information, safe. During the authentication phase, the Prover proves possession of the credential to the Relying Party, usually in an interactive way.

Our interest is focused in methods categorized as cryptographic authentication, which provide the strongest guarantees and are achieved by proving the possession and control of a cryptographic key via an authentication protocol [44].

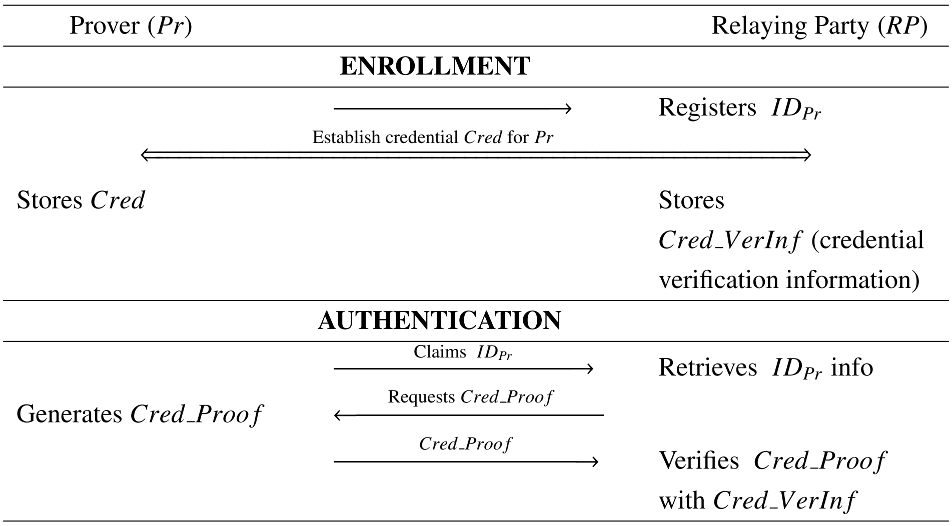
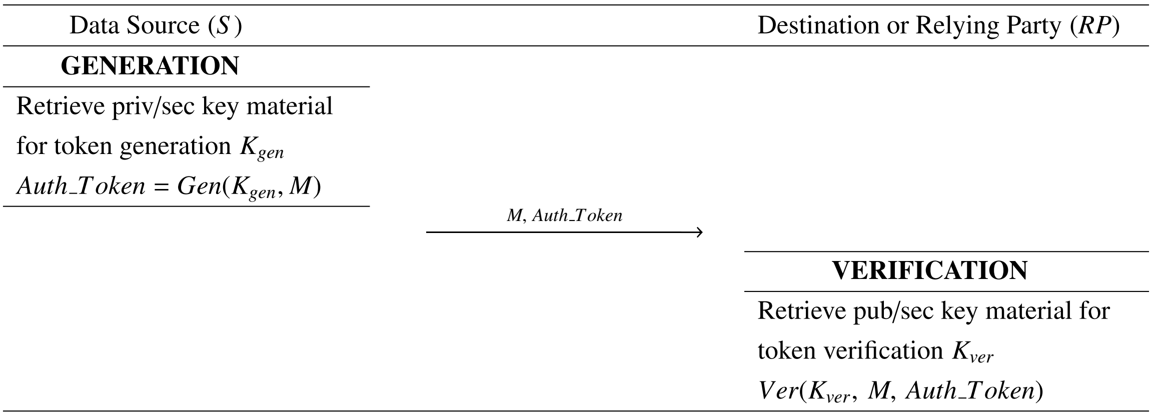


Figure 1. Entity authentication.

Data authentication. This security service involves the satisfaction of two properties: *Integrity authentication*, assurance that data has not been altered in an unauthorized manner since it was created, transmitted, or stored, and *source authentication*, assurance about the source of the data, sometimes also called origin authentication [43]. All common solutions for data authentication rely on cryptographic constructions.

Data authentication involves two phases: Generation of the authentication token and verification of such token (see Figure 2). Although there are cases where the token is embedded in the data being authenticated, the typical approach is to attach to the data M being authenticated an authentication token $Auth\_Token$ that can be verified by the Relying Party RP, i.e., the entity receiving or retrieving the data.

The authentication token is generated by the entity acting as the Data Source S using K_{gen}, a secret key k shared with the Relying Party, if symmetric cryptographic schemes are used, or a private key sk only known to the Data Source, if asymmetric cryptographic schemes are used. During verification, the Relying Party uses K_{ver}, the symmetric key k shared with the data source or the public key pk of the Data Source, together with the data M being authenticated to verify the authentication token $Auth\_Token$. If the verification is successful, the data is accepted; if the verification fails, the data is rejected. In both scenarios, symmetric or asymmetric, the authentication process operates under the assumption that the keys involved are legitimate, thereby ensuring that the Relying Party can trust the identity of the Data Source. Here, identification should be interpreted in a broad sense; for instance, in two-party or multi-party symmetric data authentication, it may be sufficient to establish that only authorized or trusted entities possess the shared key, without requiring additional explicit evidence of identity.



Data Source (S)		Destination or Relying Party (RP)
GENERATION		
Retrieve priv/sec key material for token generation K_{gen} $Auth\_Token = Gen(K_{gen}, M)$		
	$\xrightarrow{M,\ Auth\_Token}$	
		VERIFICATION
		Retrieve pub/sec key material for token verification K_{ver} $Ver(K_{ver}, M, Auth\_Token)$

Figure 2. Data authentication.

Data authentication services can also be used to provide entity authentication. As mentioned, cryptographic entity authentication is based on the claimant (Prover) proving knowledge and possession of a secret or private key. In order to do so, in the authentication phase, the Relying Party sends to the Prover a fresh unpredictable nonce. Then, the Prover sends back the authentication token generated over the nonce.

Major cryptographic tools for authentication. Today, the major cryptographic tools used for data authentication are message authentication codes (MAC) and digital signature schemes (DSS). We give formal definitions and further discuss these two tools in Sections 3 and 4. The authentication tokens that are generated and attached to the authenticated data are called authentication tags in the first case and signatures in the second. MAC constructions, usually combined or integrated with symmetric encryption algorithms to form authenticated encryption (AE) schemes, are widely used to protect all types of data transmitted or stored. Where possible, MACs are the first choice because of their reliability and high performance in terms of both computation time and storage size. On the other hand, digital signatures are not usually chosen to authenticate data because their performance is inferior to that of MACs for the same security requirements. However, their role is critical in at least the next two use cases:

- Public key authentication. A trusted entity binds an identity to a public key by signing both pieces of data (among other data) and generating what is known as a public key certificate.
- Data authentication with legally binding requirements. Digital signatures, supported by public key certificates that meet specific national or international requirements, are recognised by many laws around the world (for example, the European Digital Identity Framework [45]).

As far as it goes, all the cryptographic authentication methods rely on at least one of the entities, the claimant (or claimed Prover), securely keeping a secret and using it in the authentication phase or in the generation of the authentication token. At present, the highest level of security guarantees for the case of entity authentication is achieved when the cryptographic basic tools mentioned above are integrated with or reinforced by hardware-based cryptographic authentication devices. In the context of human authentication, common examples of such devices include smart cards and USB tokens, which typically require a second authentication factor, such as "something you know" (e.g., a password) or "something you are" (e.g., a biometric characteristic). However, when the entity we want to authenticate is itself a device (e.g., an IoT sensor), the scenario gets significantly more challenging, there is no possibility to use the aforementioned second-factor authentication methods and it is assumed a priori that the adversary can have physical access to the device. The difficulties are greater when the scenario involves devices with limited capabilities, such as in the Internet of Things, Connected Industry, or Smart Cities. In recent decades, so called physical unclonable functions (PUFs) have been proposed as a new hardware-based security primitive that has the potential to solve or at least overcome some of the challenges in device authentication. Some nice overviews on hardware security that highlight the critical role of (PUFs) in constrained device authentication can be found in [46, 47]. We further discuss this hardware-based tool in Section 5.

Authentication of cryptographic keys. Finally, a particularly critical application of data authentication is that concerning shared cryptographic keys. Although we do not delve into the underlying techniques in this survey, it is worth highlighting their relevance in the post-quantum context and providing some references for the interested readers.

Indeed, ensuring the authenticity of keying material is essential, as the inability to confirm which entities possess specific keys significantly undermines security, regardless of the cryptographic mechanisms those keys are intended to protect. Two crucial components in this context are authenticated key establishment (AKE) protocols and key encapsulation mechanisms (KEMs). AKEs are cryptographic constructions through which $n \geq 2$ users establish a shared secret key, typically communicating through an insecure channel. To this aim, during the protocol execution users typically authenticate using a priori shared secrets, either high entropy (e.g., certified cryptographic keys) or low entropy (passwords).

In turn, KEMs are public key cryptographic protocols through which a sender is able to generate and transfer an encapsulated secure cryptographic key to a dedicated receiver, holding the secret key that matches the public key used for the encapsulation. As we mentioned in the introduction, these tools are one of the major actors in the NIST process for post-quantum standarization, as they can be used as a building block for secure public key encryption and also for two party AKE. Indeed, there are several techniques for building two party AKE from KEMs which are proven secure in a post-quantum world (see, for instance, [48, 49]). When more than two users come into play, achieving authenticated post-quantum group key establishment is also possible combining two party tools in relatively simple ways (e.g., [50, 51]).

2.3. Authentication in practice

This section contains some examples of real systems in which the cryptographic tools mentioned above are used for the authentication of entities and data.

The best-known and most widely used distributed system that illustrates the usage of authentication tools is the Internet, specifically, the Transport Layer Security (TLS) protocol [52]. In the more common configuration of TLS 1.3, the handshake protocol implements a one-way AKE between the client (e.g., a web browser) and the server (e.g., a web server), where the client authenticates the server by verifying the server's X.509 public key certificate [53]. DSS are used extensively; for instance, the public key certificates are typically signed by the issuer and the server signs some of the sent messages during the handshake. Supported signature schemes in TLS version 1.3 are variants of RSA [54] and DSA over elliptic curves (ECDSA [55] and EdDSA [56]). On the other hand, the key exchange protocol recommended in TLS 1.3 is the ephemeral Diffie-Hellman over finite field groups or over elliptic curve groups, with specific variants defined in TLS 1.3 [52].

Other messages in the handshake are authenticated using a MAC. After finishing the key exchange, both parties derive additional cryptographic material from the shared key. Some of the derived keys are used to encrypt and authenticate all exchanged data using an AE scheme with additional data (AEAD) [57].

Although the previous approach is not the only one considered in network security (e.g., DNSSEC [58] uses only digital signatures and Kerberos [59], in its traditional deployment, uses only symmetric key cryptography), there are quite a number of security protocols for the Internet that are designed in a similar way to TLS, i.e., trust is established through the use of public key certificates, then, an AKE protocol is executed, and, finally, data is securely exchanged using a MAC or an AE scheme. Other examples of protocols that follow this approach are IPSec [60]—used for example in virtual private networks (VPN), or the Signal protocol [61], used in messaging applications to provide end-to-end encryption and authentication—.

In scenarios where the devices involved have limited capabilities, such as Machine-to-Machine (M2M) or IoT systems, it may be necessary to limit the options considered in the approaches discussed previously or to adopt different ones. Some M2M or IoT communication protocols rely on TLS (or DTLS [62]) to encrypt and authenticate the data exchanged between peers at the application layer. However, in highly constrained scenarios, AKEs based on public key certificates are not always possible and establishing trust through a pre-shared key is the most appropriate or only feasible approach. Many protocols (e.g., TLS and DTLS, and specially those designed for contrained devices) include the possibility to derive the cryptographic material needed to secure the exchanged data using a pre-shared key. This pre-shared key can be embedded in the device at manufacture or generated at bootstrapping by direct (secure) communication with the server or by derivation from a seed or password (e.g., printed on the device packaging). Examples where it is typical to use pre-shared keys include MQTT [63] over TLS, CoAP [64] over DTLS, and CoAP with OSCORE [65].

Embedding or storing the cryptographic material in the device may be too risky in certain use cases, since it can be assumed that once the device is deployed, an adversary may eventually gain access to it and attempt to break the device's protections to obtain the cryptographic material inside. In these scenarios, PUF-based solutions are very promising. PUF-based security hardware has been known in academia since 2002 [66] and has recently attracted an increasing attention from standards bodies [67, 68] and industry. Some examples of commercial products that have successfully achieved the PSA Certified Security Assurance Certificate are the Intrinsic ID QuiddiKey 300[†] and the PUFsecurity PUFcc Crypto Coprocessor[‡]. Both hardware systems offer a secure Root-of-Trust chip with a PUF at its heart, which can be used to provide device authentication with the highest guarantees, especially in embedded systems, and, to execute a AKE protocol. From then on, data exchanged between the parties

[†] https://products.psacertified.org/products/quiddikey-300

[‡] https://products.psacertified.org/products/pufcc

can be authenticated and encrypted using an AE scheme.

Focusing now on human authentication, NIST Digital Identity Guidelines [44] establish that the permitted authenticator types to achieve the highest assurance level (AAL3) are single-factor cryptographic authenticators combined with a password or multi-factor cryptographic authenticators. Cryptographic authenticators, either software or hardware, use an authentication protocol to prove possession and control of a cryptographic key. A typical example of this type of authenticator in recent decades has been the cryptographic smart cards used as national electronic identity documents (eIDs). These cards typically contain a pair of asymmetric keys and the routines necessary to use them securely: one pair is used to authenticate the citizen (by requiring them to sign a piece of data generated for that authentication in a way that is transparent to the user) and the other is used to sign documents.

Today, the concept of digital identity has evolved and digital wallets, either software or hardware, are seen as a suitable mechanism to store a variety of a user's credentials and the functionality to prove their ownership. A prominent example of a digital wallet is the European Digital Identity (EUDI) Wallet [69], which is currently being developed and rolled out. As expected, EUDI Wallets support user authentication that relies on digital signatures but will also support other advanced cryptographic protocols to prove possesion of a credential in more private ways or selective disclosure of attributes (e.g., with Zero-Kowledge Proofs).

Another very relevant example of cryptographic authenticators for humans are security keys. They rely on public key cryptography (through digital signatures) to help users authenticate themselves in online services. They are seen as the successor to password-based authentication and are a key technology for achieving secure, scalable and more private passwordless authentication. A popular choice for systems supporting security keys is the FIDO2 set of open authentication standards from the Fast IDentity Online (FIDO) Alliance [70].

2.4. *A prominent use case: Authentication in QKD*

Quantum key distribution (QKD) protocols exploit the features of quantum mechanics to enable the transport of a cryptographic key through an insecure (quantum) channel. While QKD protocols represent the most significant practical advancement in incorporating quantum technologies into security, it is far from being a replacement for many use kcases where classical key agreement protocols are currently used (see [71]). In particular, QKD protocols require authenticated channels, which to date is only achieved through traditional (non-quantum) tools.

Typically, in a QKD protocol quantum states are exchanged or transported through an insecure channel and measured upon reception. Thereafter, an authentication phase is implemented to assert that the assumed peer has indeed been the actual sender; most often this step is cried out by classical means (for instance, by using pre-shared keys and MACs). The complexity of such a setup indeed increases for QKD networks involving multiple users; in a naive setting, we would need $\frac{n(n-1)}{2}$ two-party keys to be able to connect n users in pairs. Another approach is to employ post-quantum digital signature schemes; however, this method has a significant drawback, as it relies on computational assumptions and necessitates a form of public key infrastructure to manage the distribution, updating, and certification of keys. For related experimental results, see [72].

In the literature, there are also proposals for building authentication leveraging quantum technologies. For instance, in [73] the authors put forward a protocol for QKD network systems using authenticated quantum states to get entity authentication. That is, certain qubits that are encoded with pre-shared information are generated and exchanged to verify the legitimacy of each entity. This other proposal from Farré et al. [74] exploits entanglement to attain user-server authentication in the context of QKD. In this survey, we focus on authentication attained through classical means, therefore, readers interested in quantum entity authentication protocols are referred to the recent report [12].

3. Symmetric key data authentication

For many years, Grover's search algorithm [2] has been considered the only realistic quantum algorithm threatening the security of symmetric cryptographic tools. The robustness of these tools depends on the secrecy of a key, with the speed of an exhaustive search over the corresponding key space commonly used as a benchmark for their strength. As Grover provides quadratic speed up in exhaustive search attacks, dodging this menance seemed simple; if keys are encoded as bitstrings, just doubling the key size would do the job. However, later attacks like that of Kuwakado and Hidenori [75] evidenced that considering brute force attacks on the key space may not be enough. In Kuwakado and Hidenori's work they consider adversaries that beyond local quantum computations may furthermore query different related (e.g., encryption) oracles in superposition. Similar superposition attacks have been proposed since then and clearly evidence that further dedicated analysis of symmetric tools in models is needed (see, for instance, [76–79]).

Still, symmetric tools for authentication are widely used and believed to work reasonably in a post-quantum context; we briefly discuss here the most relevant examples.

3.1. Message authentication codes

As mentioned already, message authentication codes (MACs) are symmetric primitives used to preserve the integrity of messages by creating an authentication tag computed by combining the message and the secret key; this tag can later be verified, thus proving that the message has not been tampered with, by those holding that same secret key.

Formally [80], a MAC is defined as a triplet of algorithms $\mathsf{MAC} = (\mathsf{KeyGen}, \mathsf{Tag}, \mathsf{Vf})$ (see Figure 3) where:

- The probabilistic key generation algorithm $\mathsf{KeyGen}(1^\ell)$ takes as input the security parameter and outputs a key $k \in \{0, 1\}^{p(\ell)}$ for a suitable polynomial $p(\ell)$.[§]
- The probabilistic authentication algorithm $\mathsf{Tag}(k, M)$ takes as input a key k, and a message M, and outputs a tag t.
- The deterministic verification algorithm $\mathsf{Vf}(k, M, t)$ takes as input a key k, a message M, and a tag t and outputs a decision: 1 (accept) or 0 (reject).

Classically, a MAC system is said to be secure if no polynomial time adversary Adv can win the following game: A random key k is chosen from the key space and the attacker is presented with a $\mathsf{MAC}(k, *)$ oracle from which he can obtain (polynomially) many tags for messages of his choice. Typically, Adv wins the game if it can produce, with non-negligible probability, what is called an existential forgery, that is, a new valid message-tag pair (M, t), where t is not the output of an oracle

[§] In the sequel, for the sake of simplicity, we will assume $\mathsf{KeyGen}(1^\ell)$ actually selects k uniformly at random in $\{0, 1\}^{p(\ell)}$.

query on input *M*. This security notion is called *existential unforgeability under chosen message attacks* (EUF-CMA).

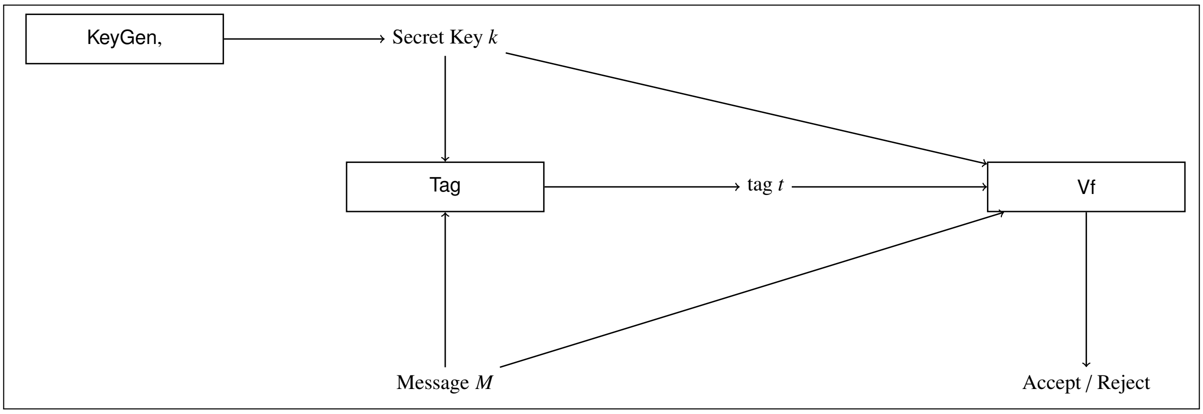


Figure 3. Sketch of MAC schemes.

One of the first security models for proving security of MACs in a post-quantum context is the one by Boneh and Zhandry [81], which introduce *indistinguishability under quantum chosen-message attacks* (EUF-qCMA) in order to mimic the classical EUF-CMA and capture superposition queries. This model (often referred to as BZ) is further refined in several papers, such as [82], where the notion *blind unforgeability* (BU) is put forward. The idea behind this notion is to rule out attacks in which the adversary has access to a MAC oracle on a certain region and uses it to produce tags for messages on a disjoint message region. For quantum adversaries who are able to make queries in superposition, these attacks are nor covered by the definition from [81] (which, moreover, did not quite take into account the destructiveness of quantum meassurement).

In [82], it is proven that several natural MAC constructions are secure in this sense, in particular:

- Quantum secure pseudorandom functions; this allows for the construction of secure variable-length MACs from any quantum secure PRF.
- Secure hash and MAC constructions; if one combines a blind unforgable hash with a so-called Bernoulli-preserving hash (see [82] for precise definitions).

Thus, there are nice choices for MAC constructions that can be used with very good security guarantees in presence of quantum adversaries; concrete examples are given in [82]. We recommend the Master thesis [83] for a nice introduction to the topic.

3.2. Authenticated encryption

Authenticated encryption (AE) schemes are cryptographic constructions aiming at providing both confidentiality and authentication. They typically combine a symmetric encryption scheme and an message authentication code (or other authentication mechanism), often following a simple sequencial/paralel strategy (Encrypt-then-MAC, Encrypt-and-MAC, MAC-then-Encrypt, etc.).

Recall that symmetric encryption schemes are defined as a tuple of algorithms (KeyGen, Enc, Dec) as follows:

- A probabilistic key generation algorithm $\mathsf{KeyGen}(1^\ell)$ that takes a security parameter ℓ and outputs a secret key k.
- A probabilistic encryption algorithm $\mathsf{Enc}(k, m)$ that takes a secret key k and a plaintext message m from a message space $\mathcal{M}$, and outputs a ciphertext c.
- A deterministic decryption algorithm $\mathsf{Dec}(k, c)$: that takes the secret key k and a ciphertext c and outputs either the original plaintext message m or a special failure symbol $\perp$ if decryption fails.

Since some symmetric encryption schemes process only small-sized message blocks, they must be applied repeatedly to encrypt large data files. These repeated applications can be organized in various structures, commonly referred to as modes of operation. Often, we speak of AE modes, which are specialized modes of operation that enable us to implement secure AE schemes from suitable symmetric encryption schemes. That is, for instance, the case of the famous Galois Counter Mode (GCM) that builds on the well-known counter mode for encryption to attain data authentication (see [84]).

In [85], Bellare and Nampempre introduced security notions in the classical setting and further analized generic composition methods for the design of these tools. In a post-quantum setting, a similar study is carried over in [86]. There, a security model is proposed, combining the classical Bellare-Namprempre security model with the ideas from [87] to handle message authentication against quantum adversaries. Furthermore, a generic construction method is given, combining any quantum-resistant symmetric-key encryption scheme together with any authentication scheme (DSS or MAC) admitting a classical security reduction to a quantum-computationally hard problem. At this, the security level required for the symmetric encryption scheme is called IND-qCPA, which is similar to the classical notion capturing so-called *indistinguishability under chosen plaintex attacks* and where challenge messages can only be encrypted classically.

For the MAC scheme, the security level required is that from [81].

A more recent exploration of security models and generic constructions for quantum-resistant AE can be found in [88]. There, it is proven that certain classical AE modes do not attain the desired security level, for instance, generic SIV mode, GCM mode or EAX mode. The attacks relevant here are quantum period finding attacks, see [79]. There are, however, also secure construction techniques identified in this work, such as the nonce-prefix variant of CBC-MAC described by [89], if implemented from a secure enough[¶] block cipher. The authors of [88] also introduce a new technique, nonce-based re-keying, to upgrade a a weakly secure authenticated encryption scheme into a secure one.

Besides generic constructions, there are also concrete specialized authenticated encryption modes of operation that have been explored in the post-quantum setting. A nice lightweight construction is SATURNIN [90]. Further, Jason et al. present a sponge-based construction in [91]. Also, Bhaumik et al. propose in [92] the so-called QCB mode of operation, which gives very strong guarantees (BZ secure authentication and IND-qCPA encryption) as soon as it is implemented with a quantum-secure tweakable block cipher.

[¶] In the sense of the so-called Q2 model, where the adversary is allowed to send queries in superposition to the challenger, mimicking a legitimate user whose answers may also be in superposition.

4. Digital signature schemes

Signature schemes are public key constructions involving, typically, two actors, a Signer and a Verifier. The Signer holds a secret key which allows him to build signatures, and a message-signature pair can be checked by the Verifier, who has access to the public key of the Signer. More formally, a digital signature scheme (DSS) is defined as a triplet of algorithms ($\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Vrfy}$), where:

- The probabilistic key generation algorithm, $\mathsf{KeyGen}(1^\ell)$, takes as input the security parameter ℓ and outpus a pair (pk, sk) of public and secret keys, of length polynomial in ℓ.
- The probabilistic signature generation algorithm, $\mathsf{Sign}(M, sk)$, takes a message M and a secret key sk and outputs a message-signature pair (M, σ).
- The deterministic verification algorithm,$\mathsf{Vrfy}(M, \sigma, pk)$, outputs one bit reflecting if the σ is a correct signature w.r.t. pk and M, that is,

$$\mathsf{Vrfy}(1^\ell, M, \sigma, pk) \to 1 \text{ iff } \sigma \text{ is consistent w.r.t. } M \text{ and } pk.$$

As it was the case for MACs, the security notion typically used to validate the security of a DSS is $\mathsf{EUF\text{-}CMA}$, *existential unforgeability under chosen message attacks*. At this, a challenger generates a key pair (pk, sk) and hands the verification key pk to the adversary, who has then access to a signing oracle that will sign (polynomially many) messages $M_1, \ldots, M_t$ of his choice, using the secret key sk. The choices of these messages can be made adaptively (not in one shot). Then, the adversary wins the game if he is able to output a pair (M^*, σ) for a new message M^* and such that $\mathsf{Vrfy}(1^n, M^*, \sigma, pk) \to 1$. To prove the DSS secure, one needs to asert that any efficient adversary will fail in this experiment with overwhelming probability.

4.1. The NIST selection for post-quantum DSS

As discussed, in 2016, the National Institute of Standards and Technology (NIST) initiated a competition to select post-quantum cryptographic constructions, identifying CRYSTALS-Dilithium, FALCON, and SPHINCS$^+$ as the first three digital signature schemes for standardization. The first and the last schemes have already been published as FIPS 204 and FIPS 205 standards (see ML-DSA [25] and SLH-DSA [93]), while the second one is expected to be published as FN-DSA (future FIPS 206). Note that the standards introduce some differences from the original NIST Round 3 proposals of the corresponding algorithms (for instance, the randomness generation in Dilithium has been adjusted and there are stricter input/output formats for SPHINCS$^+$). However, these differences are not fundamental and essentially result in implementation consistency and clarity.

Given that three of these schemes are based on lattice problems, and with the objective of fostering a more diverse portfolio of cryptographic options, NIST issued a call for additional signature proposals in July 2022 (see https://csrc.nist.gov/projects/pqc-dig-sig). An initial pool of 40 candidates underwent the first round of evaluation. Following a review period of just over a year, NIST announced in October 2024 that 14 proposals had advanced to the next round. These proposals are categorized into six distinct groups based on their underlying mathematical foundations: codes (CROSS, LESS), lattices (Hawk), MPC-in-the-head (Mirath, MQOM, PERK, RYDE, SDitH), multivariate (UOV, MAYO, QR-UOV, SNOVA), symmetric tools (FAEST), and isogenies (SQISign). Details on these constructions and on the selection criteria followed in this process can be found in [32].

The minimum requirement for the candidates was to significantly outperform Dilithium and FALCON (for lattice-based proposals) or SPHINCS$^+$ (for proposals based on other paradigms). Additionally, the criteria used by NIST to select the 14 candidates were, in decreasing order of relative importance: (1) Security, (2) computational costs and performance, and (3) the characteristics of the algorithm and the proposed implementation.

From a security standpoint, NIST placed high value on submissions that included a formal security proof, although this was not an absolute requirement. Additionally, NIST highlighted several desirable security properties, such as resistance to side-channel attacks, multi-key security, and robustness against misuse (e.g., misuse of randomness or secret keys). Submitters were also encouraged to emphasize extra security guarantees beyond standard unforgeability: For instance, exclusive ownership, message-bound signatures, or non-resignability, which could offer practical advantages in specific applications. In addition, each submission was required to include a summary of known cryptanalytic attacks, together with detailed complexity estimates. This made it possible to evaluate the current security landscape of each scheme in a transparent and informed way.

In terms of performance and efficiency, the evaluation took into account key metrics such as public key size, signature size, and computational cost. Schemes with clearly documented estimates for key generation, signing, and verification were viewed favorably. From an algorithmic design perspective, NIST expressed particular interest in proposals that demonstrated the ability to scale across a broad range of platform; from embedded devices to servers.

In some cases, schemes were selected not only for their security and performance, but also for their conceptual clarity or originality. NIST generally favored submissions that followed well-motivated design principles or introduced innovative paradigms. Indeed, diversity in these kind of process is crucial, as a wide variety of underlying assumptions and techniques reduces the risk that a single cryptanalytic breakthrough could compromise a large portion of standardized algorithms.

In the next section, we outline key ideas behind a selection of signature schemes from this shortlist or closely related ones.

4.2. The quest for post-quantum DSS

The introduction outlined key mathematical areas relevant to post-quantum cryptography. This section explores specific problems within these areas and how they are used to construct signature schemes, with a particular focus on the constructions from the NIST standardization process.

Lattice-based signatures. A nice and intuitive example of a lattice-based DSS is that of FALCON [15]. It follows a design first introduced by Gentry, Peikert and Vaikuntanathan [94], typically referred to as the GPV construction (see Figure 4). In a nutshell, given the security parameter, the key generation algorithm will select a long basis for a lattice (namely, a matrix $A \in \mathbb{Z}_q^{n\times m}$ with integer entries so that lattice elements can be generated as $A^T s \bmod q$ for $s \in \mathbb{Z}^n$). Now, the private key will be a short basis for the same lattice. Note that here, "long" and "short" refer to the length (norm) of involved vectors; shorter vector basis allow for computing solutions for the related lattice problems (CVP and SVP) efficently, while longer basis will enable only the efficient verification of a given solution to this problem.

With these ingredients, signing algorithm Sign is designed as follows:

- It generates a random value r.
- It hashes the message together with r yielding $c = H(M\|s)$.
- It uses the short basis to solve the CVP in the lattice, thus finding a lattice point v close to c.
- It defines $\sigma = (\sigma_1, \sigma_2) = (r, c - v)$.

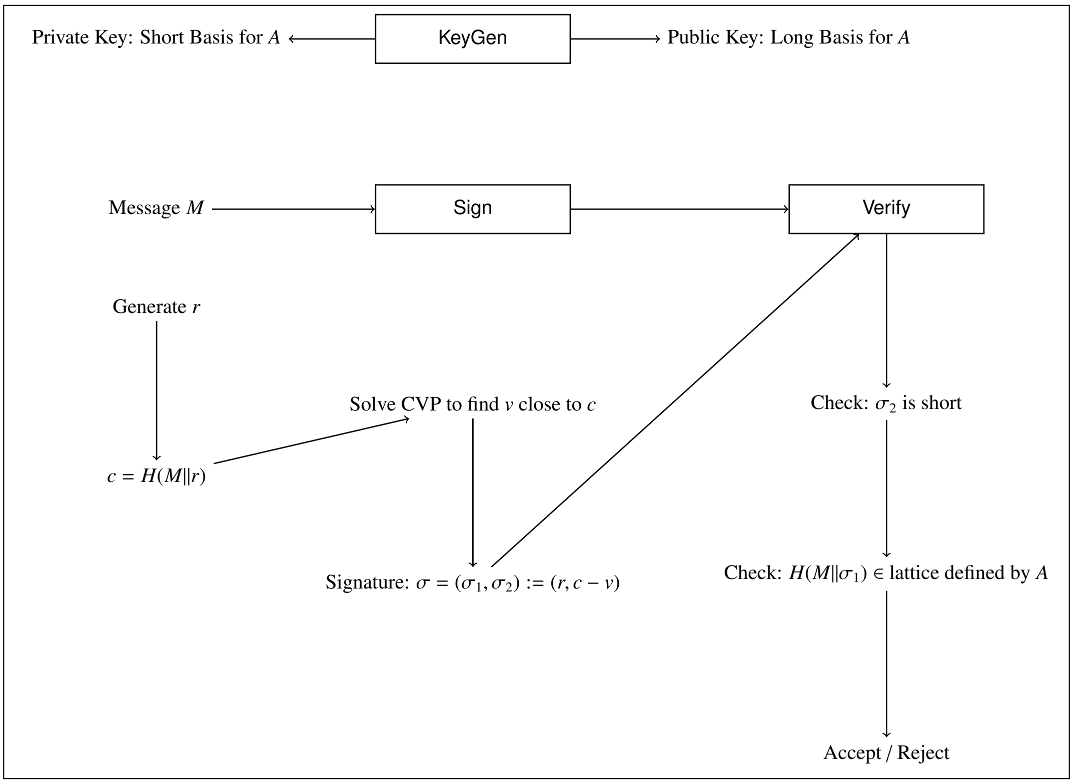

Private Key: Short Basis for A
KeyGen
Public Key: Long Basis for A
Message M
Sign
Verify
Generate r
Solve CVP to find v close to c
Check: σ_2 is short
$c = H(M\|r)$
Signature: $\sigma = (\sigma_1, \sigma_2) := (r, c - v)$
Check: $H(M\|\sigma_1) \in$ lattice defined by A
Accept / Reject

Figure 4. Sketch of GVP signatures.

Now, the verification algorithm on input (M, σ) and the public basis (A) for the lattice, will accept only the signature as correct if two conditions are met:

- σ_2 is *short.*
- $H(M\|\sigma_1)$ is a point in the lattice defined by A.

While it is clear that this construction heavily relies on the hardness of the CVP problem on the underlying lattice, there are also a number of subtle points on the design that makes FALCON not only secure, but also very efficient. In particular, the lattices chosen for generating the keys are so-called NTRU-lattices, which are moreover encoded using polynomials, and the solution to the CVP needed in the signature process is implemented through a very efficient technique called Fast Fourier Sampling. We refer to the FALCON project web page for details [15].

Code-based signatures. Here we briefly describe LESS, a code-based signature scheme presented by Baldi et al. [95], which has been selected to advance to Round 2 in the NIST signature competition.

LESS is constructed using the so called Fiat-Shamir Transform [96], which has proven extremely useful towards the design of post-quantum signature schemes. Amos Fiat and Adi Shamir introduced this technique back in Crypto 1986, as a generic method of building signature schemes from interactive proofs of knowledge. This, in particular, enables us to derive a signature scheme from any identification scheme built as a proof of knowledge of a private key (see Figure 5).

Prover		Verifier
Input:sk		Input:pk
	$\xrightarrow{init}$	
		CH u.a.r
	$\xleftarrow{CH}$	
	$\xrightarrow{RSP}$	
		$Output := OUT(pk, init, CH, RSP)$

Figure 5. Cryptographic identification scheme.

In a nutshell, the idea is to use as a base an identification scheme (see Figure 5) that can be seen as a proof of knowledge for a secret key matching a public one. At this, the Prover is presented with a random challenge CH, from which he needs to construct a response RSP that is consistent with his "declared" public key *pk* and the whole transcript of his communication with the Verifier. The Verifier will only accept if RSP cannot be constructed without the secret key that matches *pk*.

Then, the idea of the Fiat-Shamir transform is to use the message to be signed as a challenge so that the Signer proofs knowledge of a secret key using the underlying identification scheme. The proof of knowledge (RSP) would then constitute the signature, univoquely linked to both the message and the secret key (see a sketch in Figure 6).

Signer (pk, sk)		Verifier (V)
$CH := H(Init\|\|m)$		
$\sigma := RSP$	$\xrightarrow{(m,\sigma)}$	verifies consistency of $(pk, Init, CH, RSP)$

Figure 6. Fiat-Shamir transform.

For the sake of simplicity, we describe here the LESS Identification Scheme, from which the signature scheme is derived using the Fiat-Shamir Transform.

Once the security level aimed at is fixed, some public parameters are fixed and publitized; three numbers $q, n, k \in \mathbb{N}$, a matrix $G \in \mathbb{F}_q^{k\times n}$ and a hash function H. Now, the key generation process follows a common strategy within Code-Based Cryptography; the public key is a matrix that generates a "seemingly random code" while the secret key consists of a few matrices transforming the public key into a code with a fast decoding algorithm. Namely, it selects an invertible $k \times k$ matrix S and a monomial $n \times n$ matrix Q, both with entries from $\mathbb{F}_q$. These two matrices are the secret key, while the public key is $G' = SGQ$. Now, the identification interaction, through which the Verifier asserts that the Prover holds the secret key, goes as described in Figure 7.

A few caveats: In the scheme above we have omitted the fact that the hash is applied to matrices only after they have been written in a special way, systematic form, so that exactly one generator matrix is linked to each code. Further, in order to have a small cheating probability, the above procedure needs to be iterated, e.g., the prover will generate t commitments, receive t challenges from the Verifier and should answer back with t responses. The iterated protocol is then turned into a signature scheme using the Fiat-Shamir Transform, thus yielding the LESS DSS (see [95]).

Finally, let us briefly explain the (decoding) problem underlying the security of LESS, the so called *Linear Equivalence Problem* or LEP. Let G_o and G_1 be two $k \times n$ (systematic) generator matrices over the finite field $\mathbb{F}_q$, then find Q, a monomial matrix ($n \times n$ matrix with exactly one non-zero element in each row and column) so that G_1 is the reduced row echelon form of G_0Q. This is a particular case of the *Code Equivalence problem* (CEP) we introduced in Section 2.

Prover		Verifier
Input: $sk = (S, Q)$		Input: $pk = G'$
$\hat{Q} \xleftarrow{\$} \mathbb{F}_q^{n\times n}$		
$\hat{G} := G\hat{Q}$		
$h := H(\hat{G})$		
.	$\xrightarrow{h}$	
		$b \xleftarrow{\$} \{0, 1\}$
	$\xleftarrow{b}$	
$\mu := (Q^{-1})^b\hat{Q}$		
	$\xrightarrow{\mu}$	
		if $b = 0$ and $H(G\mu) = h$
		or $b = 1$ and $H(G'\mu) = h$ set $Output := 1$.
		else $Output := 0$.

Figure 7. LESS identification scheme.

Multivariate signatures. In a nutshell, most cryptographic constructions in the area are based in the hardness of a problem that we may formalize as explained next. To construct a key pair whose relationship is explicitly based on the MQ problem, the following strategy is typically used: The secret key is an easily invertible quadratic function $\mathcal{F} : \mathbb{F}^n \mapsto \mathbb{F}^n$, known as the central map. The "nice" structure of $\mathcal{F}$ is concealed using two invertible affine transformations

$$\mathcal{S} : \mathbb{F}^m \mapsto \mathbb{F}^m \text{ and } \mathcal{T} : \mathbb{F}^n \mapsto \mathbb{F}^n.$$

The public key is then constructed as a set of polynomials (explicitly defining the adversary's target system) via function composition:

$$\mathcal{P} = \mathcal{S} \circ \mathcal{F} \circ \mathcal{T} : \mathbb{F}^n \mapsto \mathbb{F}^m,$$

where the three mappings $\mathcal{S}$, $\mathcal{F}$ and $\mathcal{T}$ constitute the secret key of the scheme. The major strategies for solving this problem correspond to attack vectors, which can be either generic (such as solving the system of equations directly using Gröbner bases) or specific to a given $\mathcal{F}$ (so-called structural attacks).

With these ingredients, a typical so called oil and vinegar (UOV) DSS is depicted as follows: Let $\mathbb{F}$ be a finite field, and let o and v be two fixed integers. We define, for $n = o + v$, the two index sets $V = \{1, \dots, v\}$ and $O = \{v + 1, \dots, n\}$. Consider the polynomial ring $\mathbb{F}(x_1, \dots, x_n)$ where we refer to variables *vinegar* those with an index in V, (i.e., x_i with $i \in V$), and to variables *oil* those with an index in O. If $o = v$ the construction is called balanced (OV), and otherwise unbalanced (UOV).

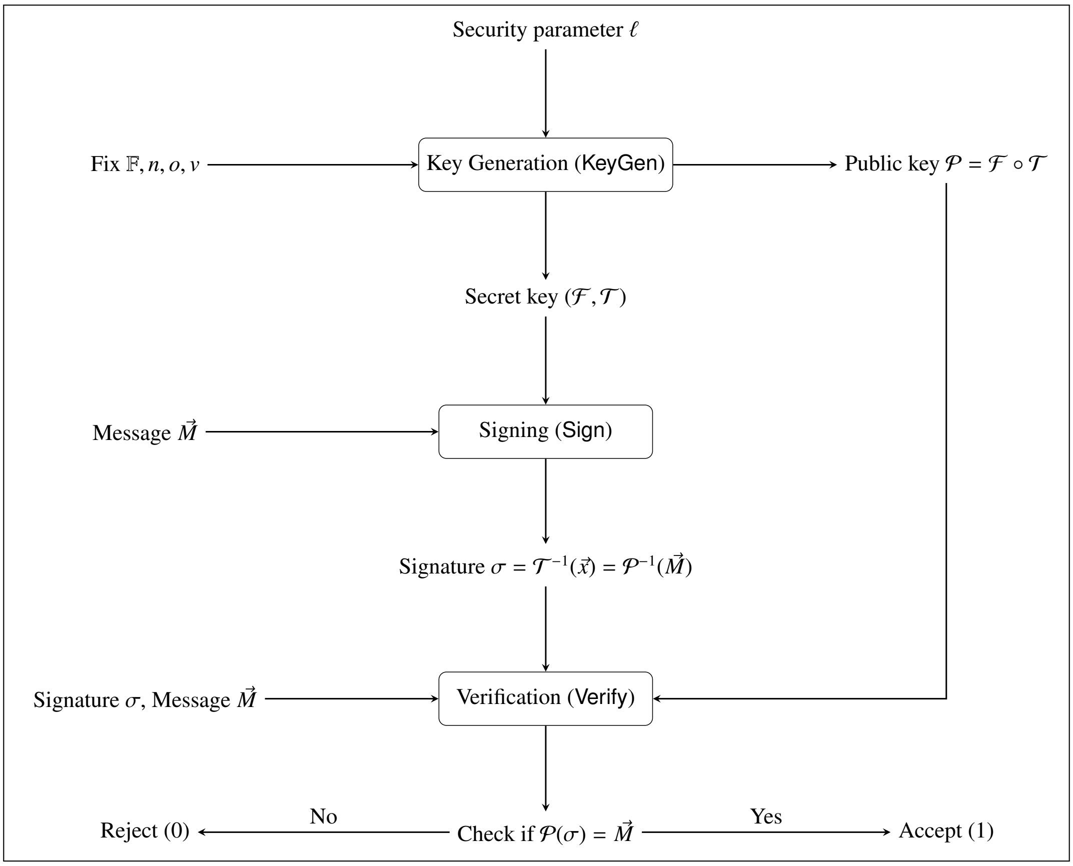

Security parameter ℓ
Fix $\mathbb{F}, n, o, v$
Key Generation (KeyGen)
Public key $\mathcal{P} = \mathcal{F} \circ \mathcal{T}$
Secret key $(\mathcal{F}, \mathcal{T})$
Message $\vec{M}$
Signing (Sign)
Signature $\sigma = \mathcal{T}^{-1}(\vec{x}) = \mathcal{P}^{-1}(\vec{M})$
Signature σ, Message $\vec{M}$
Verification (Verify)
Reject (0)
No
Check if $\mathcal{P}(\sigma) = \vec{M}$
Yes
Accept (1)

Figure 8. Sketch of UOV signatures.

Now, a sketch of a resulting signature scheme can be seen in Figure 8 and it is explained next:

- The key generation algorithm, KenGen, takes as input the security parameter ℓ chooses the so-called central application, $\mathcal{F} : \mathbb{F}^n \mapsto \mathbb{F}^n$, fixing o quadratic polynomials of the form

$$f^k(x_1, \dots, x_n) = \sum_{i,j \in V} \alpha^k_{ij} x_i x_j + \sum_{i \in V, j \in O} \beta^k_{ij} x_i x_j + \sum_{i \in O \cup V} \gamma^\ell_i x_i + \delta^\ell$$

  with

$$\alpha^k_{ij}, \beta^k_{ij}, \gamma^k_i, \delta^k \in \mathbb{F}, \forall i, j \in \{1, \dots, n\}, \forall k = 1, \dots, o.$$

  Additionally, an affine map $\mathcal{T} : \mathbb{F}^n \mapsto \mathbb{F}^n$ is chosen, with $\mathcal{P} = \mathcal{F} \circ \mathcal{T}$ being the public key and the pair $(\mathcal{F}, \mathcal{T})$ the secret key —in the UOV design, the second affine map $\mathcal{S}$ is omitted.
- The signature algorithm, Sign, is described as follows: To sign a message M, the first step is to encode it as a vector $\vec{M} \in \mathbb{F}^o$ (through a hash function), then a pre-image of $\vec{M}$ is calculated under

$\mathcal{F}$, which we call $\vec{x} \in \mathbb{F}^n$, and the signature is computed as $\sigma = \mathcal{T}^{-1}(\vec{x}) \in \mathbb{F}^n$. The calculation of $\vec{x}$ is straightforward due to the structure of the polynomials that define $\mathcal{F}$: random values are assigned to the "vinegar" variables, $x_1, \ldots, x_v$, and substituted into the polynomials f^k for $k = 1, \ldots, o$. Thus, o linear polynomials in the oil variables, $x_{v+1}, \ldots, x_n$, are obtained, which define a linear system that can be easily solved using Gaussian elimination. If the system is inconsistent, the process is repeated with a new selection of values for the "vinegar" variables. This process is straightforward because the polynomials $f^1, \ldots, f^o$ do not have any quadratic terms in the "oil" variables.

- The verification algorithm, Verify, receives a pair (d, σ), computes the vector $\vec{M}$, and checks if it matches $\mathcal{P}(\sigma)$. If so, it accepts the verification (returns 1), otherwise, it returns 0.

As we mentioned, the main disadvantage of such schemes is the difficulty of properly estimating the complexity of the underlying mathematical problem for key generation. Nonetheless, they present a promising alternative to current schemes, especially for designing digital signature methods, as it is evidenced by the fact that as many as four (out of 14) proposals for post-quantum signatures in the second round of the NIST competition are based on this paradigm (namely, UOV [97], MAYO [98], QR-UOV [99], and SNOVA [100]).

Isogeny-based signatures. Galbraith, Petit and Silva presented in [101] one of the first constructions for identification and signatures using the hardness of some problems related to isogenies . Their work later inspired the design of SQISign, the only isogeny-based signature scheme that remains under evaluation in the NIST post-quantum signatures standarization process (see [38, 102]).

In their article [101], Galbraith et al. presented two identification protocols, which are again later used for deriving signature schemes through the Fiat-Shamir Transform. To illustrate how they work, we sketch next a simplified version.

Prover		Verifier
Input: $sk = \varphi$		Input: $pk = (E_0, E_1)$
		selects a random isogeny $\psi : E_1 \longrightarrow E_2$
	$\xrightarrow{E_2}$	
		$b \overset{\$}{\leftarrow} \{0, 1\}$
	$\xleftarrow{b}$	
if $b = 0$ $\mu := \varphi$		
else $\mu := \eta$		
	$\xrightarrow{\mu}$	
		if $b = 0$ and $H(G\mu) = h$
		if the isogeny is correct set $Output := 1$.
		else $Output := 0$.

Figure 9. Isogeny-based identification scheme.

The public key consists of two elliptic cuves (E_0, E_1) and the Prover holds the corresponding secret key, consisting of an isogeny $\varphi : E_0 \longrightarrow E_1$. They interact again through a process that should be iterated enough times to convince the Verifier (just as it was the case for the LESS identification scheme explained above): first, the Prover selects a random isogeny $\psi : E_1 \longrightarrow E_2$ and sends E_2 to the Verifier. Then the Verifier selects a bit b uniformly at random and sends it to the Prover. Now, if $b = 0$, the Prover reveals ψ. Otherwise, the Prover sends an isogeny $\eta : E_0 \longrightarrow E_2$. The Verifier accepts if and only if, the received isogeny is correct (see a sketch in Figure 9).

It may seem tempting to assume the isogeny η is constructed as a composition $\psi \cdot \varphi$, but a careful second thought allows to see that then the private key would be leaked. As a matter of fact, the key point behind this design idea is the use of a (highly non trivial) method due to Kohel and others [103] to construct η in a way that it leaks nothing about φ.

Hash based signatures. Buiding signature schemes using properties from cryptographic hash functions (collision resistance, harndess to compute preimages) has been done since the seminal work of Merkle [104] in the early days of public key cryptography. The main idea behind his initial construction was to build a signature scheme leveraging a one-time signature scheme (OTS) where each key must only be used to sign one message, using trees where the leaves are labeled with hash values of OTS public keys and inner nodes are labeled by the hash of the labels of its child nodes. This way, the root node can be used to authenticate all the leave nodes.

Refinements of this seminal idea have given rise to celebrated and widely used schemes, such as XMSS [105] and SPHINCS$^+$ [106] both currently standarized (in NIST SP 800-208 [107] and FIPS 205 [93], respectively). The first is a so-called stateful scheme, i.e., it requires to keep track of which signature index has been used to avoid reuse of keys, while SPHINCS$^+$ avoids this by using stateless randomization techniques (at the cost of larger signatures and slower signing).

Signatures from multiparty computation in the head. Perhaps one of the most prominent new lines of work in the NIST process for digital signature standardization is the so-called Multiparty Computation in the Head (MPCitH), introduced in 2007 by Ishai et al. [108]. Five of the fourteen schemes currently under evaluation in the NIST competition belong to this paradigm. The core idea of this paradigm is to generate zero-knowledge proofs derived from secure multiparty computation (MPC) protocols. The approach consists of simulating an MPC protocol execution offline, where a single entity plays the role of all parties, thus avoiding interaction, while proving that the execution was performed correctly in zero knowledge. Typically, the signing process involves proving knowledge of a secret key without revealing it, through a simulated MPC protocol that necessarily involves said secret key. In particular, this key may be associated either with a (more or less extended) symmetric cryptographic scheme or with a mathematical problem tied to a one-way function. The latter strategy is followed by the schemes selected by NIST in the ongoing signature competicion within this paradigm, specifically: SDitH [109] and RYDE [110] are based on Syndrome Decoding, Mirath [111] is in the so-called MinRank problem of finding the lowest-rank element in a given matrix space, MQOM [112] is in the MQ problem, and PERK [112] on the so-called Permuted Kernel Problem of finding a permutation of a known vector such that the resulting vector lies in the kernel of a given matrix).

5. PUF-based entity authentication

Physical unclonable functions (PUF) are hardware-based security primitives that are often used for establishing trust when constrained devices are involved. Their key characteristic is non-replicability due to their physical properties, meaning their security is not reliant on computational assumptions that might be compromised by emerging quantum algorithms. As a result, their potential in enabling post-quantum authentication is certainly worth investigating.

Next, a brief overview of the main properties of PUFs and how they are utilized for entity authentication is provided.

5.1. Physical unclonable functions

A PUF is a physical security primitive provided by a device D that we can interact with. It has an unknown internal state, specific and unique to the device it is instrinsically bounded. The interaction with a specific instance of a PUF, PUF_D, basically follows a Challenge-Response behaviour modelled as a (probabilistic) function $PUF_D : \{0,1\}^{\kappa_1} \to \{0,1\}^{\kappa_2}$. Next, the major properties required to PUFs are introduced, although we refer the reader to [66, 113, 114] for a more in-depth definition and discussion of these properties, as well as a description of typical implementations.

- Reliability (otherwise referred to as reproducible). Defined as the consistency and repeatability of a PUF_D's responses to a given challenge across multiple interactions. While it is generally accepted that responses should be identical for a given challenge, this is not always the case in practice. Reliability is typically evaluated by the Hamming distance (*intra-distance*) and can be specified as follows, for a negligible small ϵ_1:

$$Pr[dist(y,z) > t \mid x \overset{\$}{\leftarrow} \{0,1\}^{\kappa_1}, y \leftarrow PUF_D(x), z \leftarrow PUF_D(x)] \leq \epsilon_1.$$

- Uniqueness. Responses given by a specific PUF_D are unique to that specific device and depend on its singular physical characteristics. The responses of a PUF_D should be at a significant Hamming distance (*inter-distance*) from those given by any other PUF'_D (even from the same manufacturing batch). For a sufficiently small ϵ_2, the following condition should be met:

$$Pr[dist(y,z) \leq t \mid x \overset{\$}{\leftarrow} \{0,1\}^{\kappa_1}, y \leftarrow PUF_D(x), z \leftarrow PUF_{D'}(x)] \leq \epsilon_2.$$

- Tamper-resistance. Manipulations to the device will change its intrinsic characteristics, provoking different responses from the PUF_D as it were *de facto* a different device.
- Unforgeability. It is very hard to create a physical clone of a specific PUF_D, even having control of the manufacturing process.
- Unpredictability. Besides the property of unforgeability, PUFs are hard to characterize algorithmically. Informally, the response to an arbitrary challenge generated by a PUF_D cannot be predicted from previous Challenge-Response Pairs (CRPs) of other PUF instances or from the target PUF_D.

In most cases, a fuzzy extractor with an error-correcting code is employed to enhance the reliability of PUFs [115,116]. A fuzzy extractor has a pair of procedures, defined informally as follows: $(R, P) \leftarrow Gen(w)$, generates a response R and a helper string P from an input w, and $Rep(w', P) = R$, reproduces the response R from an input w' that is enough close to w and the helper string P. In addition, it is common to use R, the stable response of the PUF once the noise has been corrected, as input to a privacy amplification function in order to obtain a high entropy output (e.g., SHA-256).

One of the major threats that PUFs have to face is machine-learning modeling attacks, either classical or quantum, that may be more efective towards cloning the device. Although one effective countermeasure is to limit the number of Challenge-Response Pairs available to an adversary (e.g., by using cryptographic tools such as Zero Knowledge Proofs to prove knowledge of the response at protocol level), some recent works are already proposing PUFs that incorporate within their architecture modules based on post-quantum primitives (see [117] and [118] for two provably secure strong PUFs based on the LWE problem). We refer the reader to [119] for a short survey on this topic.

5.2. An example of PUF-based authentication construction

The properties of augmented PUFs, as decribed in the previous section, make them particularly suitable for entity authentication, as it eliminates (1) the need to assign a unique key to each device during the enrollment or identity provisioning, because a unique key can be derived from the intrinsic identity of each PUF, and (2) for the device to store it in memory, because it can regenerate the key on demand. A typical PUF-based authentication scheme involves two parties: a trusted Relying Party (RP) and peer (Pr) in possesion of a specific instance PUF_D of an augmented PUF, acting as the Prover. Informally, the protocol runs as follows (see Figure 10):

Prover (Pr)		Relaying Party (RP)
	ENROLLMENT	
$(R_i, P_i) = Gen(PUF(c_i))$	$\xleftarrow{c_i}$	$c_i \overset{\$}{\leftarrow} \{0,1\}^k$
	$\xrightarrow{(R_i,P_i)}$	Stores (ID_{Pr}, c_i, R_i, P_i)
	AUTHENTICATION	
$R'_i = Rep(PUF(c_i), P_i)$	$\xleftarrow{(c_i,P_i)}$	Selects one of the stored CRPs
	$\xrightarrow{R'_i}$	Verifies $R_i \overset{?}{=} R'_i$
		Removes record containing c_i

Figure 10. Typical PUF-based authentication.

- Enrollment. Using a secure channel, RP collects from Pr a sufficiently large set of CRPs and stores them. To collect one CRP, RP choses a random challenge c_i and sends it to Pr. Pr uses PUF_D to obtain the pair $(R_i, P_i) = Gen(PUF_D(c_i))$, that corresponds to the stable response R_i for challenge c_i and the helper string P_i needed to regenerate the response without noise. RP stores (Pr, c_i, R_i, P_i).
- Authentication. RP randomly selects an unused challenge c_i and sends it to Pr together with the helper string P_i. Pr uses the augmented PUF_D to generate an output $w'_i = PUF_D(c_i)$ and then regenerates the stable response $R'_i = Rep(w'_i, P_i)$, sending it back to RP. RP compares the response R'_i with R_i, the response obtained during the enrollment. If both responses are equal, Pr has proved that possesses the authentic device PUF_D. The entry in the database associated with

c_i, the used challenge, is deleted.

PUF-based authentication is particularly suitable for constrained devices, and recent proposals take advantage of it to build secure mutual entity authentication and key agreement protocols for Internet of Things, wireless sensor networks or smart grids [120]. This is an active area of research and examples of more recent proposals are [121–123].

6. Conclusions

While the exact impact of quantum technologies on our current authentication systems remains uncertain, several alternatives show promise in providing resilience against such challenges. In this paper, we have identified and discussed three key resources: symmetric key authentication, post-quantum public key digital signature schemes, and physical unclonable functions. Further research is necessary to determine the precise quantum resistance of these tools and the most effective way to utilize them, whether individually or in combination, across different application scenarios. In moving towards quantum-resistant solutions, it is crucial to understand the specific requirements for each application and carefully evaluate the cost-benefit trade-offs involved in transitioning to alternative authentication methods.

In Table 1, we give a simple comparative summary of these three resources, summarizing the discussion of this paper. We concentrate on four high-level features: The requirement for specialized hardware, the reliance on a public key infrastructure for managing public keys, the foundational assumptions involved (typically related to computational hardness), and overall efficiency. Naturally, a more in-depth analysis is necessary when evaluating specific tools for particular application contexts.

Table 1. High level comparison of the discussed authentication mechanisms.

Feature	Symmetric key Auth.	Post-quantum DSS	PUF-based Auth.
Needs Specialized hardware	No	No	Yes (requires a PUF circuit)
Needs PKI	No (but requires secure key distribution)	Yes	No
Assumptions	Security of symmetric primitives (e.g., AES, HMAC)	Hard problems over lattices, codes, or multivariate polynomials	Physical unclonability and uniqueness of hardware
Efficiency	High (low computation and bandwidth)	Medium to low (depends on scheme and parameters)	High (very efficient once PUF hardware is present)

Finally, the guidance provided by global standardization processes, such as those organized by NIST, is invaluable, as they encourage the participation of a wide range of researchers and practitioners.

Author contributions

Ana I. González-Tablas, María Isabel González Vasco: Conceptualization, Methodology, Investigation, Writing—original draft, Writing—review & editing. All authors contributed equally to this work, and have approved the final version of the manuscript.

Acknowledgments

MIG Vasco is supported by NATO SPS project G5985 and by grant PID2023-151238OA-I00 funded by MCIU/AEI/FEDER UE, Spain. Both authors would like to acknowledge project Vital-IoT funded by INCIBE (Ministry of Digital Transformation and Public Function) and the European Union NextGenerationEU in the framework of the Recovery and Resilience Facility (RRF). AI González-Tablas is also partially supported by grant PID2022-140126OB-I00 of the Spanish AEI.

References

1. P. W. Shor, Algorithms for quantum computation: Discrete logarithms and factoring, *Proceedings 35th Annual Symposium on Foundations of Computer Science*, 1994, 124–134. https://doi.org/10.1109/SFCS.1994.365700
2. L. K. Grover, A fast quantum mechanical algorithm for database search, *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, 1996, 212–219. https://doi.org/10.1145/237814.237866
3. V. Gheorghiu, M. Mosca, Quantum resource estimation for large scale quantum algorithms, *Future Gener. Comput. Syst.*, **162** (2025), 107480.
4. C. Portmann, R. Renner, Security in quantum cryptography, *Rev. Mod. Phys.*, **94** (2022), 025008, https://doi.org/10.1103/RevModPhys.94.025008
5. P. R. Babu, S. A. Kumar, A. G. Reddy, A. K. Das, Quantum secure authentication and key agreement protocols for iot-enabled applications: A comprehensive survey and open challenges, *Comput. Sci. Rev.*, **54** (2024), 100676. https://doi.org/10.1016/j.cosrev.2024.100676
6. N. Alnahawi, J. Muller, J. Oupický, A. Wiesmaier, A comprehensive survey on post-quantum TLS, *IACR Commun. Cryptol.*, **1** (2024). https://doi.org/10.62056/ahee0iuc
7. M. Curty, D. J. Santos, Quantum authentication of classical messages, *Phys. Rev. A*, **64** (2001), 062309. https://doi.org/10.1103/PhysRevA.64.062309
8. M. Curty, D. J. Santos, E. Pérez, P. García-Fernández, Qubit authentication, *Phys. Rev. A*, **66** (2002), 022301. https://doi.org/10.1016/S0378-4371(02)01059-2
9. H. Barnum, C. Crépeau, D. Gottesman, A. Smith, A. Tapp, Authentication of quantum messages, *The 43rd Annual IEEE Symp. on Foundations of Computer Science* IEEE, 2002, 449–458. https://doi.org/10.1109/SFCS.2002.1181969
10. M.-S. Kang, Y.-H. Choi, Y.-S. Kim, Y.-W. Cho, S.-Y. Lee, S.-W. Han, et al., Quantum message authentication scheme based on remote state preparation, *Phys. Scripta*, **93** (2018), 115102. https://doi.org/10.1088/1402-4896/aae1a1

11. M. Ziatdinov, From graphs to keyed quantum hash functions, *Lobachevskii J. Math.*, **37** (2016), 705–712. https://doi.org/10.1134/S1995080216060202

12. A. Dutta, A. Pathak, A short review on quantum identity authentication protocols: How would bob know that he is talking with alice?, *Quantum Inf. Proc.*, **21** (2022), 369. https://doi.org/10.1007/s11128-022-03717-0

13. J. W. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, et al., CRYSTALS-Kyber: A CCA-Secure Module-Lattice-Based KEM, *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*, 2018, 353–367. https://doi.org/10.1109/EuroSP.2018.00032

14. L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, et al., CRYSTALS-Dilithium: A Lattice-Based Digital Signature Scheme, *IACR Trans. Cryptographic Hardware Embedded Syst.*, **2018** (2018), 238–268. https://doi.org/10.13154/tches.v2018.i1.238-268

15. P.-A. Fouque, T. Prest, T. Ricosset, T. Pornin, P. Kirchner, V. Lyubashevsky, et al., Falcon: Fast-Fourier Lattice-Based Compact Signatures over NTRU, Submission to the NIST's post-quantum cryptography standardization process, 2018. Available from: https://falcon-sign.info/.

16. D. J. Bernstein, C. Dobraunig, M. Eichlseder, S.-L. Gazdag, A. Hülsing, P. Kampanakis, et al., SPHINCS+: Submission to the NIST Post-Quantum Project, Submission to the NIST's post-quantum cryptography standardization process, 2019. Available from: https://sphincs.org/.

17. G. Alagic, M. Bros, P. Ciadoux, D. Cooper, Q. Dang, T. Dang, et al., Status Report on the Fourth Round of the NIST Post-Quantum Cryptography Standardization Process, Technical Report NIST IR 8545, National Institute of Standards and Technology, 2025. https://doi.org/10.6028/NIST.IR.8545

18. M. Ajtai, Generating hard instances of lattice problems (extended abstract), *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing*, 1996, 99–108. https://doi.org/10.1145/237814.237838

19. O. Regev, On lattices, learning with errors, random linear codes, and cryptography, *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, 84–93. https://doi.org/10.1145/1060590.1060603

20. V. Lyubashevsky, C. Peikert, O. Regev, On ideal lattices and learning with errors over rings, *JACM*, **60** (2013), 43. https://doi.org/10.1145/2535925

21. D. Balbás, The hardness of LWE and ring-LWE: A survey, 2021. Available from: https://eprint.iacr.org/2021/1358.

22. J. Hoffstein, J. Pipher, J. H. Silverman, NTRU: A ring-based public key cryptosystem, *Algorithmic Number Theory, Third International Symposium*, 1998, 267–288. https://doi.org/10.1007/BFb0054868

23. Y. Chen, Q. Liu, M. Zhandry, Quantum algorithms for variants of average-case lattice problems via filtering, *Advances in Cryptology – EUROCRYPT 2022*, 2022, 372–401. https://doi.org/10.1007/978-3-031-07082-2_14

24. National Institute of Standards and Technology (NIST), Module-Lattice-Based Key-Encapsulation Mechanism Standard, 2024. https://doi.org/10.6028/NIST.FIPS.203

25. National Institute of Standards and Technology (NIST), Module-Lattice-Based Digital Signature Standard, 2024. https://doi.org/10.6028/NIST.FIPS.204

26. R. J. McEliece, A public-key cryptosystem based on algebraic coding theory, *DSN Prog. Rep.*, **42** (1978), 114–116. https://doi.org/10.1007/0-387-23483-7_248

27. P. S. L. M. Barreto, R. Lindner, P. Longa, M. Naehrig, J. E. Ricardini, G. Zanon, BIKE: Bit flipping key encapsulation, Submission to the NIST Post-Quantum Cryptography Standardization Project, 2017. Available from: https://bikesuite.org/.

28. D. J. Bernstein, T. Chou, T. Lange, R. Niederhagen, C. Peters, P. Schwabe, Classic McEliece: conservative code-based cryptography, Submission to the NIST Post-Quantum Cryptography Standardization Project, 2017. Available from: https://classic.mceliece.org/.

29. P. Gaborit, O. Ruatta, J. Schrek, G. Zémor, HQC: Hybrid Quasi-Cyclic Key Encapsulation Mechanism, Submission to the NIST Post-Quantum Cryptography Standardization Project, 2017. Available from: https://pqc-hqc.org/.

30. A. Kipnis, J. Patarin, L. Goubin, Unbalanced oil and vinegar signature schemes, *International Conference on the Theory and Applications of Cryptographic Techniques*, 1999, 206–222. https://doi.org/10.1007/3-540-48910-X_15

31. J. Ding, D. Schmidt, Rainbow, a new multivariable polynomial signature scheme, *International conference on applied cryptography and network security*, 2005, 164–175. https://doi.org/10.1007/11496137_12

32. G. Alagic, M. Bros, P. Ciadoux, D. Cooper, Q. Dang, T. Dang, et al., Status Report on the First Round of the Additional Digital Signature Schemes for the NIST Post-Quantum Cryptography Standardization Process, 2024. https://doi.org/10.6028/NIST.IR.8528

33. J. H. Silverman, *The arithmetic of elliptic curves*, New York: Springer, 1986.

34. W. Beullens, L. D. Feo, S. D. Galbraith, C. Petit, Proving knowledge of isogenies: A survey, *Des. Codes Cryptogr.*, **91** (2023), 3425–3456. https://doi.org/10.1007/s10623-023-01243-3

35. R. Azarderakhsh, M. Campagna, C. Costello, L. D. Feo, B. Hess, A. Jalali, et al., Supersingular isogeny key encapsulation, Submission to the NIST Post-Quantum Cryptography Standardization Project, 2017. Available from: https://github.com/microsoft/PQCrypto-SIKE.

36. W. Castryck, T. Decru, An efficient key recovery attack on SIDH, *Advances in Cryptology - EUROCRYPT 2023*, 2023, 423–447. https://doi.org/10.1007/978-3-031-30589-4_15

37. W. Castryck, T. Lange, C. Martindale, L. Panny, J. Renes, CSIDH: An efficient post-quantum commutative group action, *Advances in Cryptology - ASIACRYPT 2018*, 2018, 395–427. https://doi.org/10.1007/978-3-030-03332-3_15

38. J. Chavez-Saab, M. C.-R. Santos, L. de Feo, J. K. Eriksen, B. Hess, D. Kohel, et al., Sqisign, Submission to the NIST Post-Quantum Signatures Standardization Project, 2023. Available from: https://sqisign.org/.

39. M. Bläser, Z. Chen, D. H. Duong, A. Joux, T. N. Nguyen, T. Plantard, et al., On digital signatures based on group actions: QROM security and ring signatures, *Post-Quantum Cryptography - 15th International Workshop*, 2024, 227–261. https://doi.org/10.1007/978-3-031-62743-9_8

40. D. Kahrobaei, V. Shpilrain, Using semidirect product of (semi) groups in public key cryptography, *Pursuit of the Universal - 12th Conference on Computability in Europe*, 2016, 132–141. https://doi.org/10.1007/978-3-319-40189-8_14

41. C. Battarbee, G. Borin, J. Brough, R. Cartor, T. Hemmert, N. Heninger, et al., On the semidirect discrete logarithm problem in finite groups, *Advances in Cryptology – ASIACRYPT 2024*, 2024, 330–357. https://doi.org/10.1007/978-981-96-0944-4_11

42. K. Horan, D. Kahrobaei, Hidden Subgroup Problem and Post-quantum Group-based Cryptography, *International Congress on Mathematical Software – ICMS 2018*, 2018, 218–226. https://doi.org/10.1007/978-3-319-96418-8_26

43. E. Barker, *Recommendation for key management: Part 1 - General*, Gaithersburg: National Institute of Standards and Technology, 2020. https://doi.org/10.6028/NIST.SP.800-57pt1r5

44. D. Temoshok, J. Fenton, Y.-Y. Choong, N. Lefkovitz, A. Regenscheid, R. Galluzzo, et al., *Digital identity guidelines: Authentication and authenticator management*, Gaithersburg: National Institute of Standards and Technology, 2024. https://doi.org/10.6028/NIST.SP.800-63b-4.2pd

45. European Commission, Regulation (EU) 2024/1183 of the European Parliament and of the Council of 11 April 2024 amending Regulation (EU) No 910/2014 as regards establishing the European Digital Identity Framework, 2024. Available from: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A32024R1183.

46. A. Cirne, P. R. Sousa, J. S. Resende, L. Antunes, Hardware security for internet of things identity assurance, *IEEE Commun. Surv. Tutorials*, **26** (2024), 1041–1079. https://doi.org/10.1109/COMST.2024.3355168

47. W. Hu, C.-H. Chang, A. Sengupta, S. Bhunia, R. Kastner, H. Li, An overview of hardware security and trust: Threats, countermeasures, and design tools, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, **40** (2020), 1010–1038. https://doi.org/10.1109/TCAD.2020.3047976

48. C. Boyd, B. de Kock, L. Millerjord, Modular design of kem-based authenticated key exchange, *Information Security and Privacy - 28th Australasian Conference*, 2023, 553–579. https://doi.org/10.1007/978-3-031-35486-1_24

49. A. Fujioka, K. Suzuki, K. Xagawa, K. Yoneyama, Practical and post-quantum authenticated key exchange from one-way secure key encapsulation mechanism, *ASIA CCS '13: Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security*, 2013, 83–94. https://doi.org/10.1145/2484313.2484323

50. J. I. Escribano Pablos, M. E. Marriaga, A. L. Pérez del Pozo, Design and implementation of a post-quantum group authenticated key exchange protocol with the liboqs library: A comparative performance analysis from classic mceliece, kyber, ntru, and saber, *IEEE Access*, **10** (2022), 120951–120983. https://doi.org/10.1109/ACCESS.2022.3222389

51. J. I. Escribano Pablos, M. I. González Vasco, Secure post-quantum group key exchange: Implementing a solution based on kyber, *IET Commun.*, **17** (2023), 758–773. https://doi.org/10.1049/cmu2.12561

52. E. Rescorla, The Transport Layer Security (TLS) Protocol Version 1.3, 2018. https://doi.org/10.17487/RFC8446

53. International Telecommunication Union (ITU-T), X.509: Information Technology - Open Systems Interconnection - The Directory: Public-key and Privilege Management Infrastructure, ITU-T Recommendation, 2021. Available from: https://www.itu.int/rec/T-REC-X.509/en.

54. K. Moriarty, B. Kaliski, J. Jonsson, A. Rusch, PKCS #1: RSA Cryptography Specifications Version 2.2, 2016. https://doi.org/10.17487/RFC8017

55. ANSI X9, Financial services - Public Key Cryptography for the Financial Services Industry - The Elliptic Curve Digital Signature Algorithm - ECDSA, 2020. Available from: https://webstore.ansi.org/standards/ascx9/ansix91422020.

56. S. Josefsson, I. Liusvaara, Edwards-Curve Digital Signature Algorithm (EdDSA), 2017. https://doi.org/10.17487/RFC8032

57. D. McGrew, An Interface and Algorithms for Authenticated Encryption, 2008. https://doi.org/10.17487/RFC5116

58. S. Rose, M. Larson, D. Massey, R. Austein, R. Arends, DNS Security Introduction and Requirements, 2005. https://doi.org/10.17487/RFC4033

59. D. C. Neuman, S. Hartman, K. Raeburn, T. Yu, The Kerberos Network Authentication Service (V5), 2005. https://doi.org/10.17487/RFC4120

60. K. Seo, S. Kent, Security Architecture for the Internet Protocol, 2005. https://doi.org/10.17487/RFC4301

61. Signal Messenger LLC, Signal specifications. Available from: https://signal.org/docs/.

62. E. Rescorla, H. Tschofenig, N. Modadugu, The Datagram Transport Layer Security (DTLS) Protocol Version 1.3, 2022. https://doi.org/10.17487/RFC9147

63. OASIS, MQTT version 5.0, 2019, Available from: https://www.oasis-open.org/standards#mqttv5.0.

64. C. Amsuss, J. P. Mattsson, G. Selander, Constrained Application Protocol (CoAP): Echo, Request-Tag, and Token Processing, 2022, https://doi.org/10.17487/RFC9175

65. G. Selander, J. P. Mattsson, F. Palombini, L. Seitz, Object Security for Constrained RESTful Environments (OSCORE), 2019. https://doi.org/10.17487/RFC8613

66. R. Pappu, B. Recht, J. Taylor, N. Gershenfeld, Physical one-way functions, *Science*, **297** (2002), 2026–2030. https://doi.org/10.1126/science.1074376

67. ISO/IEC, Information security, cybersecurity and privacy protection — Physically unclonable functions Part 1: Security requirements, 2020. Available from: https://www.iso.org/standard/76353.html.

68. ISO/IEC, Information security, cybersecurity and privacy protection — Physically unclonable functions Part 2: Test and evaluation methods, 2022. Available from: https://www.iso.org/standard/76354.html.

69. European Commission, EUDI Wallet Architecture and Reference Framework (ARF) v1.6.1. Available from: https://github.com/eu-digital-identity-wallet/eudi-doc-architecture-and-reference-framework/blob/main/docs/architecture-and-reference-framework-main.md.

70. FIDO Alliance, FIDO2: WebAuthn & CTAP. Available from: https://fidoalliance.org/fido2/.

71. Federal Office for Information Security (BSI), French Cybersecurity Agency ANSII, Netherlands National Communications Security Agency (NLNCSA), Swedish National Communications Security Authority and Swedish Armed Forces, Position paper on quantum key distribution, 2024. Available from: https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Crypto/Quantum_Positionspapier.html.

72. L.-J. Wang, K.-Y. Zhang, J.-Y. Wang, J. Cheng, Y.-H. Yang, S.-B. Tang, et al., Experimental authentication of quantum key distribution with post-quantum cryptography, *npj Quantum Inf.*, **7** (2021), 67. https://doi.org/10.1038/s41534-021-00400-7

73. H. Park, B. K. Park, M. K. Woo, M.-S. Kang, J.-W. Choi, J.-S. Kang, et al., Mutual entity authentication of quantum key distribution network system using authentication qubits, *EPJ Quantum Technol.*, **10** (2023), 48. https://doi.org/10.1140/epjqt/s40507-023-00205-x

74. P. J. Farré, V. Galetsky, S. Ghosh, J. Nötzel, C. Deppe, Entanglement-assisted authenticated bb84 protocol, 2025. Available from: https://arxiv.org/abs/2407.03119.

75. H. Kuwakado, M. Morii, Quantum distinguisher between the 3-round feistel cipher and the random permutation, *2010 IEEE International Symposium on Information Theory*, 2010, 2682–2685. https://doi.org/10.1109/ISIT.2010.5513654

76. M. Naya-Plasencia, Post-quantum symmetric cryptography, In: *Symmetric Cryptography 2: Cryptanalysis and Future Directions*, Great Britain, United States: ISTE Ltd, John Wiley & Sons, 203–213. https://doi.org/10.1002/9781394256327.ch17

77. X. Bonnetain, A. Hosoyamada, M. Naya-Plasencia, Y. Sasaki, A. Schrottenloher, Quantum attacks without superposition queries: The offline simon's algorithm, *Advances in Cryptology – ASIACRYPT 2019*, 2019, 552–583. https://doi.org/10.1007/978-3-030-34578-5_20

78. X. Bonnetain, G. Leurent, M. Naya-Plasencia, A. Schrottenloher, Quantum linearization attacks, *Advances in Cryptology – ASIACRYPT 2021*, 2021.

79. M. Kaplan, G. Leurent, A. Leverrier, M. Naya-Plasencia, Breaking symmetric cryptosystems using quantum period finding, *Advances in Cryptology - CRYPTO 2016*, 2016, 207–237. https://doi.org/10.1007/978-3-662-53008-5_8

80. Y. Dodis, E. Kiltz, K. Pietrzak, D. Wichs, Message authentication, revisited, *Advances in Cryptology - EUROCRYPT 2012*, 2012, 355–374. https://doi.org/10.1007/978-3-642-29011-4_22

81. D. Boneh, M. Zhandry, Quantum-secure message authentication codes, *Advances in Cryptology– EUROCRYPT 2013*, 2013, 592–608. https://doi.org/10.1007/978-3-642-38348-9_35

82. G. Alagic, C. Majenz, A. Russell, F. Song, Quantum-access-secure message authentication via blind-unforgeability, *Advances in Cryptology – EUROCRYPT 2020*, 2020, 788–817. https://doi.org/10.1007/978-3-030-45727-3_27

83. J. Nguyen, *Provably Quantum-secure Message Authentication Code*, PhD thesis, Karlsruher Institut für Technologie (KIT), 2022.

84. D. A. McGrew, J. Viega, The security and performance of the galois/counter mode (GCM) of operation, *Progress in Cryptology - INDOCRYPT 2004*, 2004, 343–355. https://doi.org/10.1007/978-3-540-30556-9_27

85. M. Bellare, C. Namprempre, Authenticated encryption: Relations among notions and analysis of the generic composition paradigm, *J. Cryptology*, **21** (2008), 469–491. https://doi.org/10.1007/s00145-008-9026-x

86. V. Soukharev, D. Jao, S. Seshadri, Post-quantum security models for authenticated encryption, *Post-Quantum Cryptography: 7th International Workshop, PQCrypto 2016*, 2016, 64–78. https://doi.org/10.1007/978-3-319-29360-8_5

87. D. Boneh, M. Zhandry, Secure signatures and chosen ciphertext security in a quantum computing world, *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference*, 361–379. https://doi.org/10.1007/978-3-642-40084-1_21

88. N. Lang, S. Lucks, On the post-quantum security of classical authenticated encryption schemes, *Cryptology ePrint Archive*, 218. Available from: https://eprint.iacr.org/2023/218.

89. M. V. Anand, E. E. Targhi, G. N. Tabia, D. Unruh, Post-quantum security of the cbc, cfb, ofb, ctr, and XTS modes of operation, *Post-Quantum Cryptography - 7th International Workshop, PQCrypto 2016*, 2016, 44–63. https://doi.org/10.1007/978-3-319-29360-8_4

90. A. Canteaut, S. Duval, G. Leurent, M. Naya-Plasencia, L. Perrin, T. Pornin, Saturnin: A suite of lightweight symmetric algorithms for post-quantum security, *IACR Trans. Symmetric Cryptol.*, **2020** (2020), 160–207. https://doi.org/10.13154/TOSC.V2020.IS1.160-207

91. C. Janson, P. Struck, Sponge-based authenticated encryption: Security against quantum attackers, *International Conference on Post-Quantum Cryptography*, 2022, 230–259. https://doi.org/10.1007/978-3-031-17234-2_12

92. R. Bhaumik, X. Bonnetain, A. Chailloux, G. Leurent, M. Naya-Plasencia, A. Schrottenloher, et al., QCB: Efficient quantum-secure authenticated encryption, *Advances in Cryptology - ASIACRYPT 2021 - 27th International Conference on the Theory and Application of Cryptology and Information Security*, 2021, 668–698. https://doi.org/10.1007/978-3-030-92062-3_23

93. National Institute of Standards and Technology (NIST), *Stateless Hash-Based Digital Signature Standard*, 2024. https://doi.org/10.6028/NIST.FIPS.205

94. C. Gentry, C. Peikert, V. Vaikuntanathan, Trapdoors for hard lattices and new cryptographic constructions, *STOC '08: Proceedings of the fortieth annual ACM symposium on Theory of computing* , 2008, 197–206. https://doi.org/10.1145/1374376.1374407

95. M. Baldi, A. Barenghi, L. Beckwith, J.-F. Biasse, A. Esser, K. Gaj, et al., LESS: Linear Equivalence Signature Scheme, Submission to the NIST's post-quantum cryptography: Additional DSE standardization process, 2023. Available from: https://www.less-project.com/.

96. A. Fiat, A. Shamir, How to prove yourself: Practical solutions to identification and signature problems, *Advances in Cryptology - CRYPTO 86*, 1986, 186–194. https://doi.org/10.1007/3-540-47721-7_12

97. W. Beullens, M.-S. Chen, J. Ding, B. Gong, M. J. Kannwischer, J. Patarin, B.-Y. Peng, et al., Uov: Unbalanced oil and vinegar, Submission to the NIST Post-Quantum Signatures Project, 2023. Available from: https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/round-1/spec-files/UOV-spec-web.pdf.

98. W. Beullens, F. Campos, S. Celi, B. Hess, M. J. Kannwischer, Mayo, Submission to the NIST Post-Quantum Signatures Project, 2025. Available from: https://pqmayo.org/assets/specs/mayo-round2.pdf.

99. R. Akiyama, H. Furue, Y. Ikematsu, F. Hoshino, K. Kinjo, H. Kosuge, et al., Qr-uov, Submission to the NIST Post-Quantum Signatures Project, 2025. Available from: https://info.isl.ntt.co.jp/crypt/qruov/files/qruov_Spec-v2.0.pdf.

100.L.-C. Wang, C.-Y. Chou, J. Ding, Y.-L. Kuan, J. A. Leegwater, M.-S. Li, et al., Submission to the NIST Post-Quantum Signatures Project, 2025. Available from: https://snova.pqclab.org/files/SNOVA_Round2.pdf.

101.S. D. Galbraith, C. Petit, J. Silva, Identification protocols and signature schemes based on supersingular isogeny problems, *J. Cryptol.*, **33** (2020), 130–175. https://doi.org/10.1007/978-3-319-70694-8_1

102.L. D. Feo, D. Kohel, A. Leroux, C. Petit, B. Wesolowski, Sqisign: Compact post-quantum signatures from quaternions and isogenies, *Advances in Cryptology - ASIACRYPT 2020*, 2020, 64–93. https://doi.org/10.1007/978-3-030-64837-4_3

103.D. Kohel, K. E. Lauter, C. Petit, J. Tignol, On the quaternion ℓ-isogeny path problem, *LMS J. Comput. Math.*, **17** (2014), 418–432. https://doi.org/10.1112/S1461157014000151

104.R. C. Merkle, A certified digital signature, *Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference*, 1989, 218–238. https://doi.org/10.1007/0-387-34805-0_21

105.A. Hülsing, D. Butin, S.-L. Gazdag, J. Rijneveld, A. Mohaisen, XMSS: Extended merkle signature scheme, 2018. https://doi.org/10.17487/RFC8391

106.J. Aumasson, D. J. Bernstein, W. Beullens, C. Dobraunig, M. Eichlseder, S. Fluhrer, et al., Sphincs$^+$: A stateless hash-based signature scheme, *Cryptology ePrint Archive*, Available from: https://sphincs.org/data/sphincs%2B-specification.pdf, Selected for NIST PQC standardization; see specification (e.g. sphincs.org).

107.D. Cooper, D. Apon, Q. Dang, M. Davidson, M. Dworkin, C. Miller, *Recommendation for stateful hash-based signature schemes*, NIST Special Publication, 2020. https://doi.org/10.6028/NIST.SP.800-208

108.Y. Ishai, E. Kushilevitz, R. Ostrovsky, A. Sahai, Zero-knowledge proofs from secure multiparty computation, *Proceedings of the 39th Annual ACM Symposium on Theory of Computing (STOC)* ACM, 2007, 21–30. https://doi.org/10.1145/1250790.1250794

109.C. Aguilar-Melchor, N. Gama, J. Howe, A. Hülsing, D. Joseph, D. Yue, Sd-in-the-head: Syndrome-decoding-in-the-head digital signature scheme, 2022. Available from: https://eprint.iacr.org/2022/1645.

110.L. Bidoux, J. Chi-Domínguez, T. Feneuil, P. Gaborit, A. Joux, M. Rivain, A. Vin cotte, Ryde: A digital signature scheme based on rank-syndrome-decoding problem with mpc-in-the-head paradigm, 2023. Available from: https://doi.org/10.48550/arXiv.2307.08726.

111.L. Bidoux, T. Feneuil, P. Gaborit, R. Neveu, M. Rivain, Dual support decomposition in the head: Shorter signatures from rank SD and minrank, *Advances in Cryptology - ASIACRYPT 2024 - 30th International Conference on the Theory and Application of Cryptology and Information Security*, 2024, 38–69. https://doi.org/10.1007/978-981-96-0888-1_2

112.T. Feneuil, M. Rivain, Building mpc-in-the-head-based signatures from mq, minrank, rank sd and pkp, 2022. Available from: https://eprint.iacr.org/2022/1512.

113.M. Roel, Physically unclonable functions: Constructions, properties and applications, PhD thesis, Katholieke Universiteit Leuven, 2012. Available from: https://lirias.kuleuven.be/handle/123456789/353455.

114.C.-H. Chang, Y. Zheng, L. Zhang, A retrospective and a look forward: Fifteen years of physical unclonable function advancement, *IEEE Circuits Syst. Mag.*, **17** (2017), 32–62. https://doi.org/10.1109/MCAS.2017.2713305

115. Y. Dodis, R. Ostrovsky, L. Reyzin, A. Smith, Fuzzy extractors: How to generate strong keys from biometrics and other noisy data, *SIAM J. Comput.*, **38** (2008), 97–139. https://doi.org/10.1007/978-3-540-24676-3_31

116. J. Delvaux, D. Gu, D. Schellekens, I. Verbauwhede, Helper data algorithms for PUF-based key generation: Overview and analysis, *IEEE Trans. Comput.-Aided Des. Integrated Circuits Syst.*, **34** (2014), 889–902. https://doi.org/10.1109/TCAD.2014.2370531

117. Y. Wang, X. Xi, M. Orshansky, Lattice PUF: A strong physical unclonable function provably secure against machine learning attacks, *2020 IEEE International Symp. on Hardware Oriented Security and Trust (HOST)* IEEE, 2020, 273–283.

118. X. Xi, G. Li, Y. Wang, M. Orshansky, A provably secure strong PUF based on LWE: Construction and implementation, *IEEE Trans. Comput.*, **72** (2022), 346–359.

119. S. Chowdhury, A. Covic, R. Y. Acharya, S. Dupee, F. Ganji, D. Forte, Physical security in the post-quantum era: A survey on side-channel analysis, random number generators, and physically unclonable functions, *J. Cryptogr. Eng.*, **12** (2022), 267–303. https://doi.org/10.1007/S13389-021-00255-W

120. P. Mall, R. Amin, A. K. Das, M. T. Leung, K.-K. R. Choo, PUF-based authentication and key agreement protocols for IoT, WSNs, and smart grids: A comprehensive survey, *IEEE Int. Things J.*, **9** (2022), 8205–8228. https://doi.org/10.1109/JIOT.2022.3142084

121. S. Roy, D. Das, A. Mondal, M. H. Mahalat, B. Sen, B. Sikdar, PLAKE: PUF-based secure lightweight authentication and key exchange protocol for IOT, *IEEE Int. Things J.*, **10** (2022), 8547–8559. https://doi.org/10.1109/JIOT.2022.3202265

122. P. R. Babu, A. G. Reddy, B. Palaniswamy, A. K. Das, EV-PUF: Lightweight security protocol for dynamic charging system of electric vehicles using physical unclonable functions, *IEEE Trans. Network Sci. Eng.*, **9** (2022), 3791–3807. https://doi.org/10.1109/TNSE.2022.3186949

123. S.-W. Lee, M. Safkhani, Q. Le, O. H. Ahmed, M. Hosseinzadeh, A. M. Rahmani, et al., Designing secure PUF-based authentication protocols for constrained environments, *Sci. Rep.*, **13** (2023), 21702. https://doi.org/10.1038/s41598-023-48464-z